
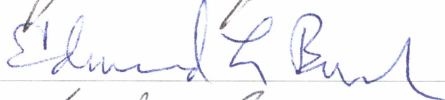



THE MEASUREMENT OF ANISOTROPIC THERMAL CONDUCTIVITY IN
SNOW WITH NEEDLE PROBES

By

Joshua Holbrook

RECOMMENDED:

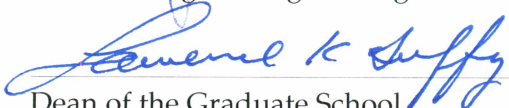




Advisory Committee Chair


Chair, Dept. of Mechanical Engineering

APPROVED:


Dean, College of Engineering and Mines


Dean of the Graduate School


Date

**THE MEASUREMENT OF ANISOTROPIC THERMAL CONDUCTIVITY IN SNOW
WITH NEEDLE PROBES**

A
THESIS

Presented to the Faculty
of the University of Alaska Fairbanks
in Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

By
Joshua Holbrook, B.S.

Fairbanks, Alaska

May 2011

Abstract

A new method for measuring thermal conductivity is being adapted from the method of measuring isotropic thermal conductivity in snow with needle probes as used by Sturm, Johnson and others, in order to enable the determination of anisotropic thermal conductivities. This method has particular relevance to measuring thermal conductivity of natural snowpacks where conductivity can be strongly anisotropic due to structures that develop from vapor transport-induced metamorphism, self-compaction and other mechanisms, and where there are known discrepancies between density-conductivity relations empirically derived from guarded hot plate and needle probe methods.

Both analytically-based solutions and finite element numerical solutions to the anisotropic case are used to calculate the expected effective thermal conductivity as a function of anisotropic thermal conductivity and needle orientation. Additionally, preliminary measurements of both anisotropic salt/sugar layered samples and of snow were taken. Both suggest that detecting anisotropy in such materials is possible, though made difficult by variability between measurements and the requirement of multiple measurements at various angles. These studies suggest that anisotropy in snow may be able to explain in part the discrepancies between guarded hot plate and needle probe measurements in certain cases.

Table of Contents

	Page
Signature Page	i
Title Page	ii
Abstract	iii
Table of Contents	iv
List of Figures	vi
List of Tables	ix
List of Appendices	x
Acknowledgements	xi
1 Introduction	1
1.1 Why Snow's Conductivity Matters	1
1.2 Thermal Conductivity Measurements of Snow	2
1.3 Snow Metamorphic Principles	3
1.4 Anisotropic Behavior in Snow	4
1.5 Motivation for Measuring Snow Anisotropy	5
1.6 Anisotropic Model	6
1.7 Document Outline	6
2 Analytical Needle Probe Approach	8
2.1 Introduction	8
2.2 The Isotropic Case	8
2.3 Difficulties in the Anisotropic Case	9
2.4 Posing The Problem in Two Coordinates	9
2.5 Coordinate Transformation	10
2.6 From Temperature Distribution to Effective Thermal Conductivity	12
2.7 Finding Effective Conductivity as a Function of Needle Orientation	13
2.8 Conclusions	13
3 Numerical Needle Probe Approach	14
3.1 Introduction	14
3.2 Geometry and Domain Properties	14
3.3 MATLAB in Geometry-Based Parametric Studies Using COMSOL 3.5a	15

3.4	Automatic Calculation of Conductivity from Simulated Time/Temperature Data	16
3.5	Convergence Study	17
3.6	Conclusions	18
4	Experimental Measurements	19
4.1	Introduction	19
4.2	Needle Probe Measurement Fundamentals	19
4.3	Snow Conductivity Measurements	20
4.4	Benchtop Tests	21
4.5	Raw Materials for the Anisotropic Composite	22
4.6	Apparatus for Containing Anisotropic Composite	23
5	Results and Interpretation	26
5.1	Parameters and Nondimensionalization	26
5.2	Numerical vs. Analytical Predictions	26
5.3	Benchtop Measurements	29
5.4	In-Situ Snow Measurements	30
5.5	Ramifications	31
6	Future Work	37
6.1	Introduction	37
6.2	Assumptions in the Analytical Approach	37
6.3	Extended Convergence Study	37
6.4	Improved Benchtop Method	38
6.5	Comprehensive Benchtop Measurements	38
6.6	Comprehensive In-Situ Measurements	39
6.7	Exploration of the Cooling Curve	39
6.8	A Method for Determining Anisotropic Thermal Conductivity From Measurements	39
7	Conclusions	41
	Bibliography	43

List of Figures

	Page
1.1 Arctic and Sub-Arctic climate is affected largely by heat transfer between the atmosphere and the ground. Snowpack adds thermal resistance transfer, affecting this heat transfer.	1
1.2 An illustration of a needle probe in cross-section. Note that the heat trace in many needle probes, including the one used in experiments for this research, actually wraps around an inner core instead of running axially through the needle.	2
1.3 Typical heating and cooling curves from a needle probe measurement. Time in the cooling curve is measured from the end of the heating curve.	3
1.4 Anisotropy in snow may either occur as a result of microstructure features, or in the aggregate due to geometry.	4
1.5 A diagram illustrating the measurement θ in models and measurements in this document. In all these cases, the angle is measured from the horizontal plane, which is also the plane of isotropy.	6
2.1 A 2-dimensional linear coordinate transformation.	10
3.1 A COMSOL screenshot showing the geometry of the finite element model, which consists of a metal needle in a sphere of a snow-like material.	14
3.2 A side-view of COMSOL's results, focusing on the needle. Colors indicate temperature.	16
3.3 An illustration of the method used to find the "long-time" slope of the numerical simulations, which used a correlation coefficient to estimate the "straightness" of a section.	17
3.4 Contents of a cell array, representing the results of a particular simulation. .	17
4.1 Extruded illustration of the needle probe apparatus. Major parts are labelled.	20
4.2 A screenshot of PC200W, the software used to pull data off the CR10X data logger.	21

4.3	A plot of temperature vs. time from a real-world measurement of snow. Each curve must be analyzed by-hand to check for such effects as convection, as seen on the right-hand side of this curve.	22
4.4	A close-up shot of tested snowpack.	23
4.5	An illustration of the apparatus used in benchtop measurements. The apparatus was designed to tilt in order to cause alternating layers of self-leveling materials to meet the needle at a given angle. However, the materials actually used were not self-leveling, meaning the tilting apparatus was of limited utility.	24
4.6	A photograph of the benchtop measurement apparatus in use, at 30 degrees.	25
5.1	A comparison of the numerical results and the analytical theory shows general agreement. Grey dots represent numerical simulation results, the grey surface represents an interpolating surface of the dots, and the blue surface represents the analytical model. Disagreement between the two may be due to edge effects and/or numerical model convergence issues.	27
5.2	Slices of theoretical predictions by angle. Black points connected by dashed lines represent numerical results, while solid blue lines represent analytical theory. It can be seen that the analytical theory predicts measured conductivity to be a stronger function of angle than the numerical data at higher conductivity ratios.	28
5.3	Slices of theoretical predictions by k_z/k_{xy} . Black points connected by dashed lines represent numerical results, while solid blue lines represent analytical theory. It can be seen that the analytical theory is perfect for the isotropic case ($k_z/k_{xy} = 1$), while the numerical experiments report larger-than-expected values.	29
5.4	A comparison of two $T(t)$ curves from equivalent simulations with different fineness of mesh. These two curves appear quite similar, but their long-time slopes are measurably different	30

5.5	Theoretical predictions for the special case of $\theta = 0$, when the needle is oriented horizontally. Blue points represent numerical solution, while green shows the line where $k_{\text{eff}} = k_z$, where measured conductivity and vertical conductivity are the same. The analytical predictions are indistinguishable from this line, and are therefore not plotted.	32
5.6	A comparison of the benchtop measurements with the numerical and analytical predictions for alternating layers of salt and sugar, given the calculated anisotropic thermal conductivity.	33
5.7	Upper and lower bounds of 95 % confidence in thermal conductivity measurements of the salt and sugar layered samples. This analysis indicates that the measurements can not statistically exclude a null hypothesis.	33
5.8	Conductivity measurements in roughly the same layer of snow at various angles. Even a limited amount of in-situ snow measurements suggest a degree of anisotropy.	35
5.9	These boxplots give a general idea of the differences in measurements between the near-horizontal angle and the more oblique ones in snow.	36
6.1	Anisotropic Conductivity Ratio vs. Material Conductivity Ratio for the layered geometry used in the benchtop experiments. It may take a very large material conductivity ratio in order to achieve a relatively minor anisotropic conductivity ratio.	38

List of Tables

	Page
3.1 Constants used in numerical models.	15
3.2 It quickly becomes impractical to increase the mesh size of a model, as increases in runtime are non-linear and are limited by both CPU and computer memory.	18
4.1 Raw results of salt and sugar measurements after calculating conductivity. The multiple results were averaged for the purpose of predicting anisotropic conductivity of an alternately-layered medium.	24
5.1 A comparison of k_{eff} from two equivalent simulations with different fineness of mesh. Despite the similarities in time/temperature curves, the resulting conductivity calculations differ by nearly 10 %. Units are in W/m·K.	31
5.2 Raw data from the benchtop measurements. Note that one of the cooling curve measurements is striked out. This is because, when examined, it is clearly an outlier. Units are in W/m·K.	34
5.3 Basic statistics on normalized benchtop measurements. Units are in W/m·K.	34
5.4 Conductivity results from the snow measurements. Units are in W/m·K.	35
5.5 Measured and derived measurements for snow density. Units are in W/m·K.	35
D.1 $\theta = [0 : 5 : 90]$, $[k_{xy}, k_z] = [0.3, 0.5]$	69
D.2 $\theta = 0$, $[k_{xy}, k_z] = \text{meshgrid}(\text{linspace}(0.2, 0.4, 4), \text{linspace}(0.2, 0.4, 4))$	70
D.3 $\theta = [30 : 15 : 90]$, $k_{xy} = 0.3$, $k_z = [0.1 : 0.1 : 0.5]$	71
D.4 $\theta = [0 : 30 : 90]$, $k_{xy} = 0.3$, $k_z = [0.1, 0.5]$	72
D.5 $\theta = [5 : 5 : 15]$, $k_{xy} = 0.3$, $k_z = [0.1, 0.5]$	72

List of Appendices

	Page
A Code Used for Chapter 2	44
A.1 model.py	44
B Code Used for Chapter 3	47
B.1 worker.m	47
B.2 mesher.m	48
B.3 solver.m	51
B.4 fitter.m	55
B.5 assembler.m	56
B.6 reFitter.m	56
B.7 analyzer.m	57
B.8 tabulator.m	60
B.9 symmetric_tocell.m	60
C Code Used for Chapter 4	62
C.1 testtools.py	62
D Raw Results of Numerical Model	69
E Needle Probe Apparatus Directions	73
E.1 Taking A Measurement	73
E.2 CSV Headers	73

Acknowledgements

Funding for this research was supplied by a grant from the Cooperative Institute For Alaska Research. Additional support in the form of HPC resources—That is, linux workstations with COMSOL—was granted by the Arctic Region Supercomputing Center.

The author would like to personally thank: His mom and dad for preparing him for adult life and for supporting him once he was out of their house; his friends and roommates at 503 Long Spur Loop for all the Magic, Munchkin and board games; his fellow engineering grad students for engaging in technical conversation; James and Peteris, for enabling productive procrastination; Donna Patrick and all the trained professionals at UAF CH&C for trying to keep him sane; Tom Kircher and the redditors at /r/statistics for their last-minute help in trying to quantify the relevance of lab experiments, and *etothepixi* from /r/askscience for helping me find bulk glycerine; Rorik Peterson, for being not only the best advisor of all time but also a friend; Jerry Johnson, Matthew Sturm, Ed Bueler and everybody else that offered their guidance and support; and his brother, for all the times he lent the use of his truck.

"No one ever told me this, but it's important in picking a research field to make sure there are people interested in it who live in nice places."

—Dr. John Rhodes

"It's non-trivial."

—Dr. Jerome Johnson, on the measurement of
snow's thermal conductivity

Chapter 1

Introduction

1.1 Why Snow's Conductivity Matters

The thermal properties of snow are of interest to scientists studying Arctic and sub-Arctic climates because, during the long, cold winters in this region, snow's thermal behavior plays a critical role in determining the net energy balance between Earth's surface and the atmosphere. After all, any heat transfer occurring between the Earth and the atmosphere over snow-covered ground must go through the snow first (Figure 1.1). In fact, the snow itself may store and release energy over time.

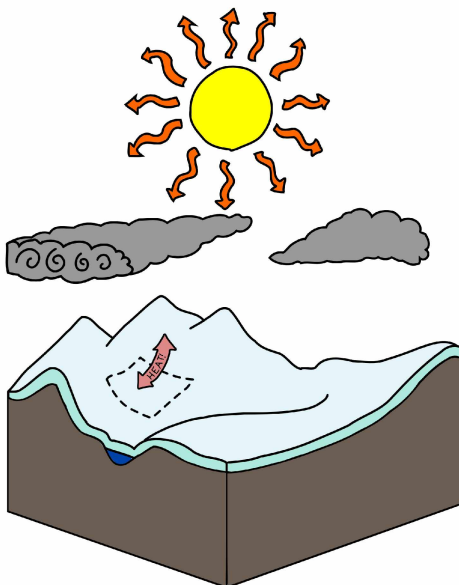


Figure 1.1: Arctic and Sub-Arctic climate is affected largely by heat transfer between the atmosphere and the ground. Snowpack adds thermal resistance transfer, affecting this heat transfer.

This energy transfer is critical in accurate climate models for these cold regions, therefore the effective thermal conductivity of snow is a very important factor for these models, and has been studied thoroughly. [Sturm *et al.*, 2002; Sturm and Johnson, 1992, 1991]

1.2 Thermal Conductivity Measurements of Snow

A typical technique for measuring thermal conductivity, especially in the context of engineering materials such as building insulation, is the guarded hot plate method. For this technique, a constant temperature gradient is induced across the material, and heat flux over the material is measured. By Fourier's Law, $k = \frac{\dot{q}l}{A\Delta T}$, where \dot{q} is the heat flux, A is the cross-sectional area of the sample, l is the sample thickness, and ΔT is the temperature difference across the sample. This technique works well in many cases.

Another technique used for porous materials, such as soils and snow, is the needle probe method. A needle probe consists of a long, thin needle with heating wire running along its interior, and a temperature sensor in the center. This configuration approximates an infinite line of constant-flux heat source (Figure 1.2). [Carslaw and Jaeger, 1959]

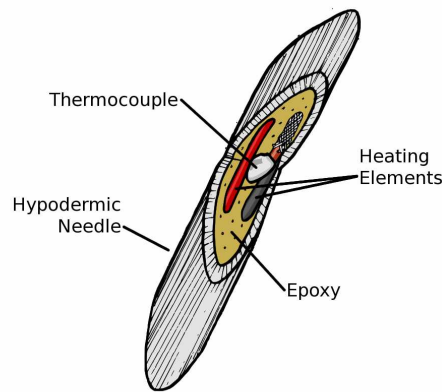


Figure 1.2: An illustration of a needle probe in cross-section. Note that the heat trace in many needle probes, including the one used in experiments for this research, actually wraps around an inner core instead of running axially through the needle.

This needle is inserted into the material whose thermal conductivity is being measured, and a constant voltage is applied to the needle's heating element. This causes a constant heat flux along the needle, and, knowing the resistance of the heat trace, this heat flux may be calculated. This causes the material's temperature near the needle to rise (Figure 1.3a). After some given amount of time, the heating element is turned off, and the temperature around the needle begins to fall back towards ambient (Figure 1.3b).

The temperature data measured over time for these two periods are called the heating cooling curves, respectively. Based on the slopes of these curves as a function of $\ln(t)$ and

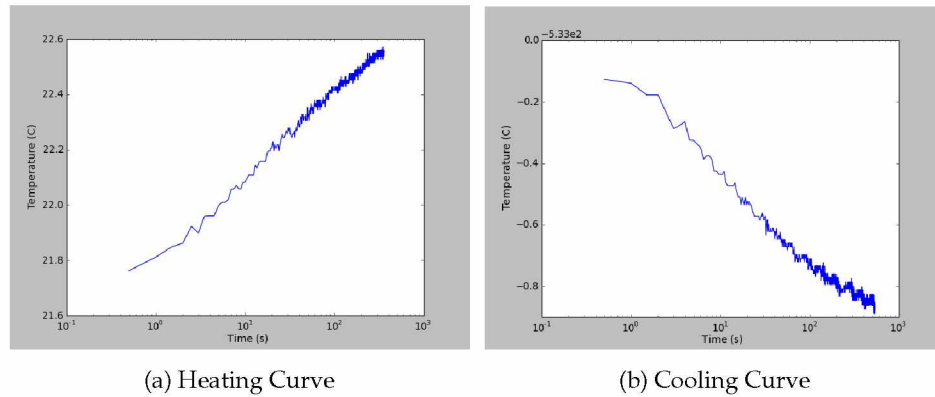


Figure 1.3: Typical heating and cooling curves from a needle probe measurement. Time in the cooling curve is measured from the end of the heating curve.

approximate analytical solutions for these situations, effective thermal conductivity may be calculated.

In this document, studies concentrate on the heating curve. In fact, the numerical and analytical approaches focus exclusively on the heating curve. However, the benchtop and in-situ measurements use both heating and cooling curves.

1.3 Snow Metamorphic Principles

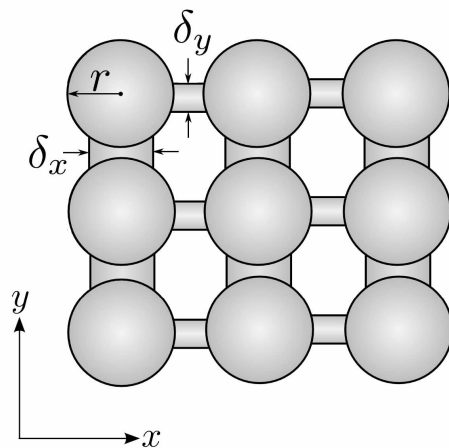
The structure of a snowpack is strongly influenced by outside environmental factors. Immediately after falling from the sky, snow begins to metamorphose as it compacts under its own weight. In addition to this, temperature gradients cause snow to sublimate and re-form in different regions of the snowpack, in a process called vapor transport. Vapor transport is best known for causing depth hoar, but occurs throughout the snowpack. Other events, such as freeze-thaw events, may also change the form of snow. All these metamorphic processes on snow cause it to form regions of varying thermal conductivity. In some cases, these regions may form sharp, distinct layers with constant properties, while in other cases they have continuously varying properties.

1.4 Anisotropic Behavior in Snow

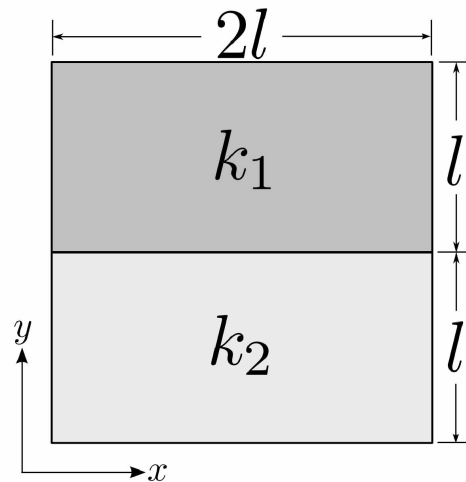
Anisotropy in snow can occur in two ways: Either due to small-scale structure in the snow, or due to macroscale features that cause anisotropy in the aggregate.

On the small scale, snow may be anisotropic due to differences in grain boundary connections, as illustrated in Figure 1.4a. [Pitman and Zuckerman, 1968] In this case, the layer of snow is itself anisotropic with respect to thermal conductivity because grains connect to each other more completely in one direction than in another. This occurs, for example, in depth hoar, where vapor transport causes the grains to develop vertically-aligned structural features.

At a macroscale, alternating regions of low-conductivity and high-conductivity material (isotropic or not) may also act in the aggregate as a single material of anisotropic thermal conductivity—In other words, the effective thermal conductivity parallel to the orientation of the layers may be different than the effective thermal conductivity orthogonal to the layers. For example, suppose a composite exists of alternating layers, each of thickness l and with conductivities k_1 and k_2 , as in Figure 1.4b.



(a) Structural anisotropy, caused by differing inter-grain boundaries.



(b) Aggregate anisotropy, caused by alternating layers of isotropic material.

Figure 1.4: Anisotropy in snow may either occur as a result of microstructure features, or in the aggregate due to geometry.

In the vertical direction, the effective thermal conductivity is $\frac{1}{2}(k_1 + k_2)$ [Lunardini, 1981]. However, in the horizontal direction the effective thermal conductivity is $2\left(\frac{1}{k_1} + \frac{1}{k_2}\right)^{-1}$. An analogous analysis could be applied to a number of geometries. Note that this particular layered geometry can not cause an aggregate anisotropy with a vertical conductivity greater than the horizontal conductivity.

1.5 Motivation for Measuring Snow Anisotropy

It is expected that thermal conductivity in snow should be correlated with snow density, since density is a function of percent composition ice and air. [Pitman and Zuckerman, 1968] This is seen in practice, based both on measurements taken with the guarded hot plate method and with the needle probe method. However, the guarded hot plate method consistently predicts higher thermal conductivity as a function of snow density than the needle probe, and nobody knows why.

One theory that could potentially explain this is that this discrepancy is caused by anisotropy in snow. Guarded hot plate methods typically measure conductivity in the vertical direction (k_z), while needle probe measurements typically measure some sort of average of vertical and horizontal thermal conductivities (k_z and k_{xy}). If k_z is consistently higher than k_{xy} in snow, then this *could* potentially explain the discrepancy.

In order to properly address this theory, it must be known if anisotropic thermal conductivity in snow can be measured with needle probes. If this can be done, then the accuracy of such determinations and the number of measurements required to make a determination, must also be known. Approaching this question is the primary focus of this research, as it enables answering the following questions:

- How severe is anisotropy in snow? Is the amount of anisotropy significant? Can horizontal measurements be used to approximate vertical thermal conductivity?
- Is anisotropy in snow predictable? That is, could one take a single measurement and extrapolate from it the anisotropic thermal conductivity?

If anisotropy in snow is significant and predictable, then anisotropy in snow may be able to explain the historical difference between guarded hot plate methods and needle probe methods. However, if snow anisotropy is typically not very severe or if needle

probe measurements should closely approximate vertical conductivities, then anisotropy in snow is likely not the explanation for these differences.

1.6 Anisotropic Model

In every model studied, it has been assumed that the horizontal plane has the same thermal conductivity and that only the vertical direction differs. In other words, $k_x = k_y = k_{xy} \neq k_z$. Each model aims to predict the effective conductivity, k_{eff} as a function of angle.

In both analytical and numerical models and in the measurements, the angle parameter, θ , is measured from the horizontal plane, as in Figure 1.5.

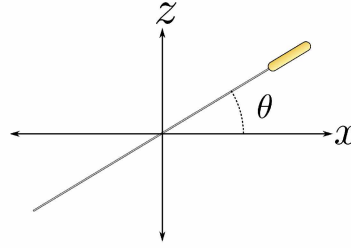


Figure 1.5: A diagram illustrating the measurement θ in models and measurements in this document. In all these cases, the angle is measured from the horizontal plane, which is also the plane of isotropy.

1.7 Document Outline

First, this document will discuss the differential equations associated with adapting the isotropic needle probe technique to the anisotropic case, as well as analytical approaches to solving them.

Second, the use of 3D finite element models in COMSOL with MATLAB to find numerical solutions to the problem will be discussed.

Then, this document will cover techniques for testing the predictions of these approaches with both real snow and engineered anisotropic materials, in this case using table salt and table sugar.

The results of the analytical and numerical approaches will then be compared to each other and to the measurements of snow and engineered materials. The meanings of these

results, and their ramifications with regards to conductivity/density relations will also be discussed.

Finally, unanswered questions and avenues for future research will be described.

Chapter 2

Analytical Needle Probe Approach

2.1 Introduction

The technique used to measure thermal conductivity with a needle probe is based on the assumption that the needle approximates an infinite line source of energy with a constant heat flux embedded in an infinite medium. The origins of the analytical solution for isotropic thermal conductivity may be found in Carslaw and Jaeger's book, "Conduction of Heat in Solids." [Carslaw and Jaeger, 1959]

The method based on Carslaw and Jaeger's solution depends on solving a 2-D problem, where all planes orthogonal to the needle have the same temperature distribution; in other words, temperature is not a function of axial position. Moreover, the problem is further simplified by posing the problem into radial coordinates and solving for temperature as a function of radial distance only.

In the isotropic case, this is straightforward, as conductivity is a constant scalar. Unfortunately the anisotropic case is more complex, but luckily not completely intractable.

2.2 The Isotropic Case

The isotropic case solves the following equation:

$$k\nabla^2 T = \rho C \frac{\partial T}{\partial t}$$

Where T is temperature, k is a scalar thermal conductivity, ρ is density, C is volumetric heat capacity, and t is time.

By casting this problem into cylindrical coordinates, the equations may be simplified such that they are a function of radial distance r only (as temperature is assumed to not be a function of axial position z or angle ϕ).

After applying this transformation and solving the equation, the analytical solution to the problem becomes:

$$T(r, t) = -\frac{q}{4\pi k} \text{Ei}\left(-\frac{r^2}{4kt}\right) \quad (2.1)$$

where q is heat flux from the needle per linear distance, and $\text{Ei}()$ is as defined in Equation 2.2 for real-valued arguments.

$$\text{Ei}(x) = - \int_{-x}^{\infty} \frac{e^{-t}}{t} dt \quad (2.2)$$

Solving for the exponential integral analytically is not possible, and numerical solutions can be difficult. Typically, an approximation for small r^2/t is used instead:

$$T(r, t) = \frac{q}{4\pi k} \ln \left(\frac{4kt}{r^2} \right) - \frac{\gamma q}{4\pi k} \quad (2.3)$$

Typical use of this function is to find $\frac{dT}{d(\ln t)}$ and solve for k . Equation 2.3 will be used for the remainder of this analysis, though it could easily be applied to Equation 2.1 as well.

2.3 Difficulties in the Anisotropic Case

The anisotropic case varies from the isotropic one in that instead of a scalar thermal conductivity k , one must solve the problem using an $n \times n$ thermal conductivity, $[K]$, where n is the number of dimensions in the problem. As a consequence, reducing the problem into two dimensions becomes more difficult. Moreover, when the problem is posed in cylindrical coordinates, the solution becomes a function not only of r , but of ϕ as well.

2.4 Posing The Problem in Two Coordinates

By assuming that temperature distribution is not a function of axial direction z , one may reduce the problem to an analogous one in orthogonal directions x and y instead:

$$\nabla_{xy} \cdot ([K]_{2 \times 2} \nabla_{xy} T) = \rho C \frac{\partial T}{\partial t} \quad (2.4)$$

Without loss of generality, it may be further simplified like so:

$$\nabla \cdot \left(\begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \nabla T \right) = \rho C \frac{\partial T}{\partial t} \quad (2.5)$$

This may be done by choosing the directions x and y such that the matrix is diagonal.

The values of k_x and k_y may be found by finding the components of $[K]$ that are in the xy plane.

In particular, Equation 2.6 was used in practice.

$$[k_x, k_y] = \text{Eig} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} [K] \right) \quad (2.6)$$

2.5 Coordinate Transformation

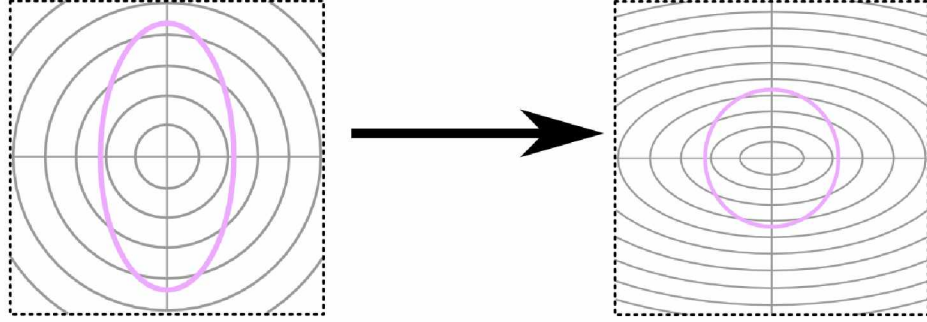


Figure 2.1: A 2-dimensional linear coordinate transformation.

In order to apply the isotropic solution to this anisotropic case, a coordinate transformation may be applied such that the problem is transformed into an isotropic case with respect to some $x' = a_x x$ and $y' = a_y y$, as in Figure 2.1. Without loss of generality, suppose $a_x = 1$.

$$\frac{dx'}{dx} = 1 \quad (2.7)$$

$$\frac{dy'}{dy} = a_y \quad (2.8)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial x'} \frac{dx'}{dx} = \frac{\partial f}{\partial x'} \quad (2.9)$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial y'} \frac{dy'}{dy} = a_y \frac{\partial f}{\partial y'} \quad (2.10)$$

$$\nabla T = \frac{\partial T}{\partial x'} \hat{e}_{x'} + a_y \frac{\partial T}{\partial y'} \hat{e}_{y'} \quad (2.11)$$

$$[K]\nabla T = k_x \frac{\partial T}{\partial x'} \hat{e}_{x'} + k_y a_y \frac{\partial T}{\partial y'} \hat{e}_{y'} \quad (2.12)$$

$$\nabla \cdot ([K]\nabla T) = k_x \frac{\partial^2 T}{\partial x'^2} + k_y a_y^2 \frac{\partial^2 T}{\partial y'^2} \quad (2.13)$$

$$(2.14)$$

Suppose the right hand side is equal to the equivalent isotropic expression:

$$k \left(\frac{\partial^2 T}{\partial x'^2} + \frac{\partial^2 T}{\partial y'^2} \right) = k_x \frac{\partial^2 T}{\partial x'^2} + k_y a_y^2 \frac{\partial^2 T}{\partial y'^2} \quad (2.15)$$

As a result,

$$k = k_x \quad (2.16)$$

$$a_y = \sqrt{\frac{k_x}{k_y}} \quad (2.17)$$

$$(2.18)$$

Therefore, the following coordinate transformation will allow for the application of isotropic solutions to an isotropic case with $k = k_x$:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{\frac{k_x}{k_y}} \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.19)$$

2.6 From Temperature Distribution to Effective Thermal Conductivity

Using Equation 2.19, the isotropic solution may be applied:

$$k_x \nabla^2 T = \rho C \frac{\partial T}{\partial t} \quad (2.20)$$

to yield the following result (for sufficiently large $t/(r')^2$):

$$T(r', t) = \frac{q}{4\pi k_x} \ln\left(\frac{4k_x t}{r'^2}\right) - \frac{\gamma q}{4\pi k_x} \quad (2.21)$$

In the isotropic case, the value of r does not change the derivative with respect to the natural log of time, as long as it is assumed constant. In contrast, the anisotropic case requires that another transformation is applied to r' so that both k_x and k_y may be recovered. This requires that some contextual meaning is assigned to r and r' . In this analysis, it is assumed that the measurement occurs at some $r = r_0$, perhaps on the surface of the physical needle.

This approach isn't without its problems. For example, it supposes that the isotherms are all circles in the transformed geometry, but if a finite-sized needle was actually being modeled in the problem then the isotherms near the (elliptically-shaped in the transformed domain) needle would be elliptical as well, and only isotherms sufficiently far away would be round. However, this approach allows us to keep using the $\ln()$ approximation, while a solution given a finite needle would likely require the use of Bessel functions.

Applying this technique to the anisotropic case, $r_{xy} = \cos(\theta)\hat{e}_x + \sin(\theta)\hat{e}_y$ must also be transformed into $r_{x'y'}$:

$$\begin{aligned} \begin{pmatrix} r_{x'} \\ r_{y'} \end{pmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{\frac{k_x}{k_y}} \end{bmatrix} \begin{pmatrix} r_0 \cos(\theta) \\ r_0 \sin(\theta) \end{pmatrix} \\ &= r_0 \left(\cos(\theta)\hat{e}_x + \sqrt{\frac{k_x}{k_y}}\hat{e}_y \right) \end{aligned}$$

$$||r'||^2 = r_0^2 \left(\cos^2(\theta) + \frac{k_x}{k_y} \sin^2(\theta) \right) \quad (2.22)$$

This means that the temperature around the needle should now vary as a function of θ , unlike in the isotropic case. Now, since the needle only measures a single value, it may be assumed that the measured quantity is an average temperature, such as the average surface temperature of the probe. This may be expressed like so:

$$T_{\text{avg}}(t) = \frac{4\pi k_x}{q} \frac{\mathcal{E}(\ln(t), \frac{k_x}{k_y})}{\mathcal{E}(1, \frac{k_x}{k_y})} \quad (2.23)$$

where:

$$\mathcal{E}(f(\phi, \alpha), \alpha) = \int_0^{2\pi} f \sqrt{\cos^2(\phi) + \alpha \sin^2(\phi)} d\phi \quad (2.24)$$

2.7 Finding Effective Conductivity as a Function of Needle Orientation

In order to extract the effective k value, a function of the form $C_1 \ln(t) + C_2$ is fit to Equation 2.23. Then, all that is left is to evaluate the functions at various combinations of $k_x y$, k_z and θ .

2.8 Conclusions

An analytical approach to studying anisotropic thermal conductivity with needle probes is more difficult than with the isotropic case. However, by using coordinate projections to pose the problem in two dimensions, and by applying coordinate transformations to the domain, one may apply the accepted isotropic theory to the anisotropic case with minimal modification. By numerically evaluating the predicted temperature distribution over time, one may find the expected conductivity measurement given the theory holds for anisotropic materials.

Chapter 3

Numerical Needle Probe Approach

3.1 Introduction

Numerical experiments were used to simulate needle probes in anisotropic mediums with three-dimensional finite element heat transfer models in COMSOL 3.5a. These consisted of a large-scale parametric study using varying combinations of k_{xy} , k_z and angle orientation θ .

3.2 Geometry and Domain Properties

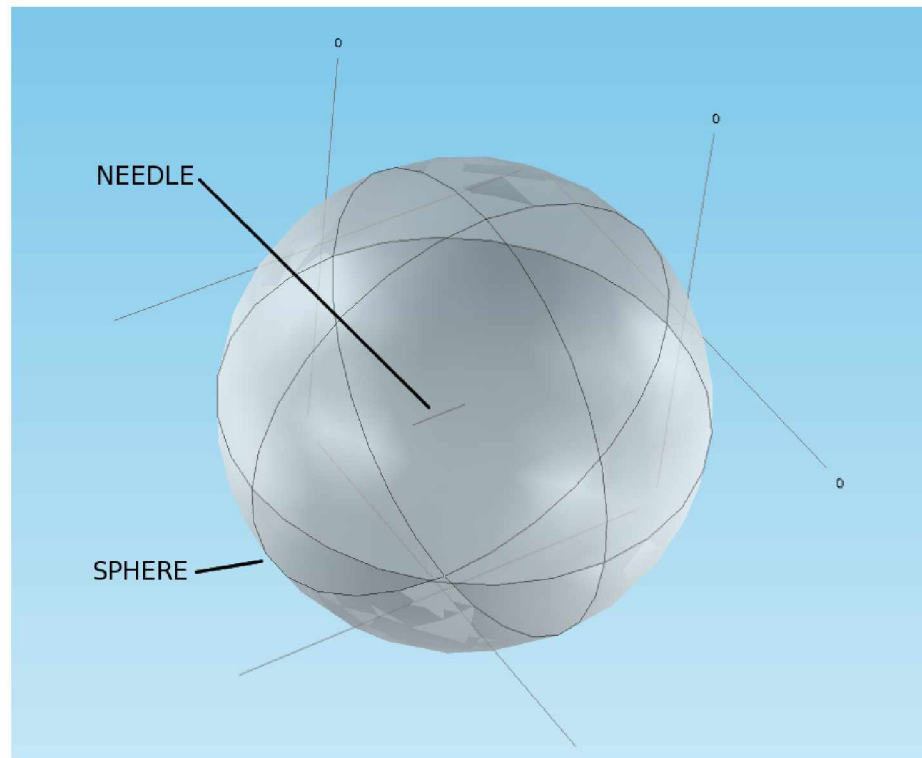


Figure 3.1: A COMSOL screenshot showing the geometry of the finite element model, which consists of a metal needle in a sphere of a snow-like material.

The needle is simulated as a long, thin steel cylinder embedded in the center of a sphere of a snow-like material (called snow in this model), as in Figure 3.1. While most of the

Table 3.1: Constants used in numerical models.

radius of needle	0.25 mm
length of needle	10 cm
radius of snow	40 cm
density of needle	8000 kg/m ³
C_P of needle	460 J/kg · K
q of needle	0.5 W/m
k of needle	160 W/m · K
density of snow	200 kg/m ³
C_P of snow	2050 J/kg · K

dimensions and material properties are held constant (see Table 3.1), the anisotropic conductivity of the snow is parameterized in the form of a 3×3 symmetric matrix with all positive eigenvalues. In practice, this is done by specifying a diagonal matrix Λ with positive eigenvalues k_{xy} and k_z and a rotation matrix R around the x axis, and then defining $K = R^T \Lambda R$ as in Equation 3.1 :

$$K = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} k_{xy} & 0 & 0 \\ 0 & k_{xy} & 0 \\ 0 & 0 & k_z \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.1)$$

The boundary conditions on the surface of the sphere enforce zero heat flux, and the radius of the sphere is chosen such that the sphere approximates an infinite medium.

Point temperatures recorded are the center of the needle, which corresponds to the location of the thermocouple used in real-world experiments, and six points on the surface of the snow, to ensure that the sphere is sufficiently large by checking for a near-zero increase in temperatures on these points. In these models, the increase in temperature is on the order of 10^{-14} degrees Kelvin.

3.3 MATLAB in Geometry-Based Parametric Studies Using COMSOL 3.5a

Unfortunately, COMSOL 3.5a does not have the facilities necessary to implement a geometry-changing multi-parameter parametric study as required from the GUI alone. However,

COMSOL 3.5a does have facilities for scripting with MATLAB.

MATLAB code written to implement the parametric study was largely auto-generated by COMSOL, by building a base model in COMSOL 3.5a and exporting to an m-file. This code is split into two parts: The meshing code, and the solving code. These pieces of code are wrapped in functions, called “mesher” and “solver” respectively, and used by a main procedure called “worker.m.”

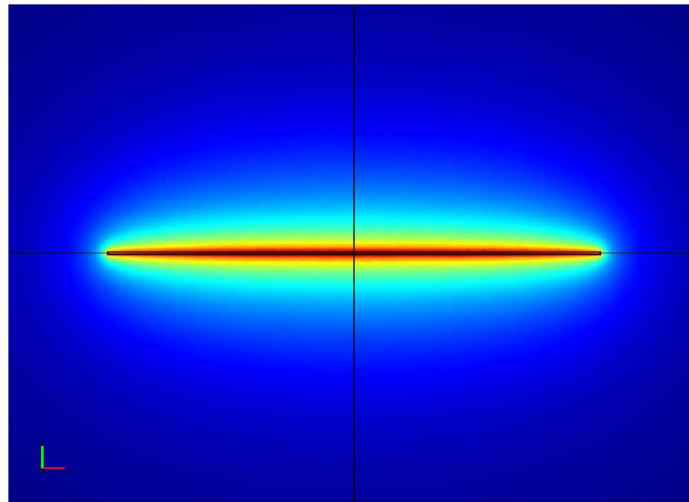


Figure 3.2: A side-view of COMSOL’s results, focusing on the needle. Colors indicate temperature.

3.4 Automatic Calculation of Conductivity from Simulated Time/Temperature Data

Results from COMSOL are automatically fitted against the linear model with respect to $\ln(t)$ by “dropping” early (t, T) datapoints until the correlation coefficient of the remaining points was sufficiently high, as in Figure 3.3. Then, a linear curvefit is applied to these remaining points. Finally, the slope of this fit is used to calculate k_{eff} .

The results from these analyses are organized into a nested cell array which mirrors the format of the two MATLAB matrices k_{xy} and k_{xz} . The data in each slot of the nested array is itself a cell array containing the simulation results, organized as shown in Figure 3.4. Typically, the only data saved from the simulation was the (t, T) data from the center of the modeled needle, the average of the temperatures on the surface of the sphere, and the

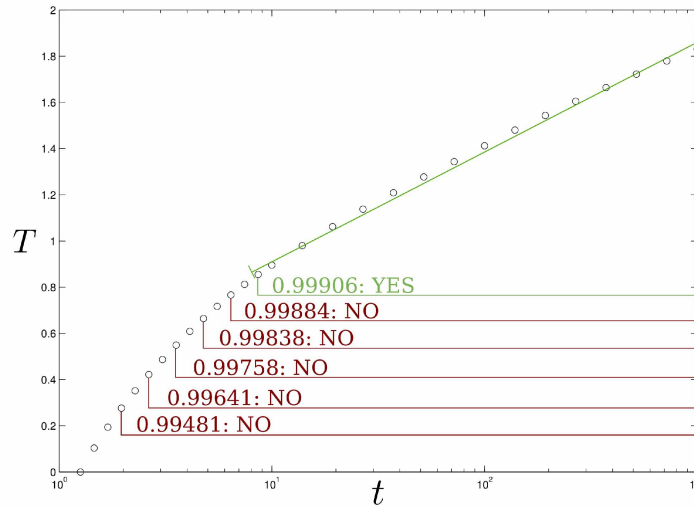


Figure 3.3: An illustration of the method used to find the “long-time” slope of the numerical simulations, which used a correlation coefficient to estimate the “straightness” of a section.

1	:	k_{meas}
2	:	[time, temperature] T
3	:	Avg. Surf. Temp.

Figure 3.4: Contents of a cell array, representing the results of a particular simulation.

simulated k_{eff} . In some runs, the raw structure representing the problem to COMSOL was exported back to the native .mph format for further study, such as viewing the temperature distribution over the entire domain as in Figure 3.2.

A number of analyses, some of more use than others, can be applied to the data. For example, one procedure applied asserted that all averaged sphere surface temperatures had increased by less than a thousandth of a degree.

3.5 Convergence Study

A small, informal convergence study was executed in order to get a general idea of how appropriate the model’s chosen grid size is. A convergence study, in this context, consists of running the same model with different grid sizes and studying how the results change as a function of grid size. The goal is to show that the problem converges on a particular

solution when grid size is sufficiently refined. While this can't prove that the solution being converged to is the *correct* solution, it can show that, given that the convergent solution is correct, that the model uses a sufficiently refined grid to get sufficiently accurate results.

For this convergence study, a particular solution was chosen from the finite element parametric study and the mesh was refined to different levels. An analysis of these results should then give an idea of the convergence properties of the problem.

One issue that occurs with such convergence studies is that refining the grid exponentially increases the difficulty of solving the resulting finite element model. In this instance, the grid could only be refined once without reaching a point where solving the problem became impractical. This gives only two data points for which to base any conclusions, but even this may be useful.

Table 3.2: It quickly becomes impractical to increase the mesh size of a model, as increases in runtime are non-linear and are limited by both CPU and computer memory.

Number of Elements	Time to Complete
35892 elements	≈ 10 minutes
159641 elements	≈ 2 hours
528945 elements	??? (> 3 weeks)

3.6 Conclusions

COMSOL 3.5a is used to run three-dimensional simulations of thermal conduction in the needle probe problem, in concert with MATLAB for automation and post-processing. The method generated useful information on predicted thermal conductivity measurements from a heating curve approach given an anisotropy and a needle orientation with respect to this needle orientation.

Chapter 4

Experimental Measurements

4.1 Introduction

In addition to theoretical results, real world cases must be measured in order to validate the theory. In this chapter, methods for making actual measurements in both snow and in engineered materials are explored. Moreover, methods for building engineered materials are discussed.

4.2 Needle Probe Measurement Fundamentals

Needle probe measurements are taken using an apparatus borrowed from the Cold Regions Research and Engineering Laboratory (CRREL), illustrated in Figure 4.1. Encased in a Pelican case for protection are a Campbell CR10X data logger, a relay switch, a 12v gel cell for the CR10X and a series of D-cells to power the heating coils in the needle. The program that came with the data logger uses the relay switch to control the heat flux from the needle, and records temperature data from the needle's thermocouple, as well as the voltage across the heating element, over the course of a five minute heating curve and a ten minute cooling curve.

Once testing is complete, data may be uploaded from the data logger using Campbell's PC200W software (shown in Figure 4.2) and a serial connection.

Using PC200W, data may be uploaded from the CR10X in a raw binary format and then converted to a .csv format. This .csv data may be analyzed with either spreadsheet software, a series of scripts, or both. For this series of experiments, Excel, Gnumeric and cat/append are used to verify the existence of data and to combine datasets, while python is used for the analysis. Generally, each analysis consists of subtracting t_0 from the relevant time intervals, subsetting the collected data over a straight section, and finding the slope. In addition, a correction factor, named the McGaw Cooling Curve Correction after a CRREL researcher, is used for benchtop measurements to account for the insulation around the apparatus.

Generally, each measurement also has some metadata associated with it. In particular, anisotropic measurements have an angle associated with them, and snow measurements also have a coordinate position on the snowpack associated with them. These are mea-

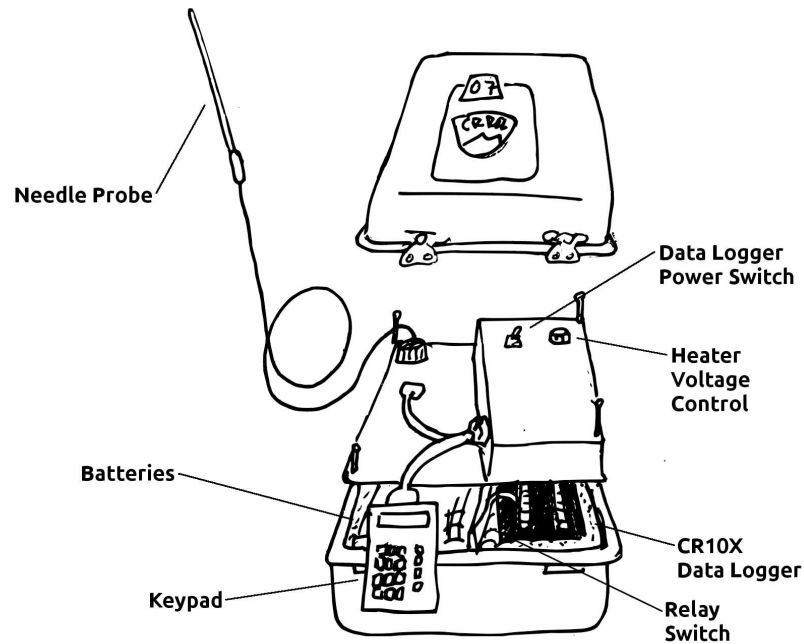


Figure 4.1: Extruded illustration of the needle probe apparatus. Major parts are labelled.

sured with a protractor and a tape measure, respectively. This means that, while taking measurements, one has to be careful to make sure that they can keep the proper metadata associated with each measurement.

Unlike the case of numerical experiments, it is extremely important in real-world experiments—especially in the case of snow—to hand-inspect every time/temperature curve. This is because, unlike the numerical experiments, there is a significant chance that data will not be usable. In the case of snow in particular, convection is typically experienced near the end of the heating curve and the beginning of the cooling curve.

4.3 Snow Conductivity Measurements

The first step in measuring proper snow is to make a vertical cut in the snowpack, as in Figure 4.4.

Then, for every measurement, the needle is inserted into snow and a measurement is taken. Snow is relatively difficult to work with due to the low structural integrity of the material. The wire connecting the probe to the data logger is stiff enough at low tempera-

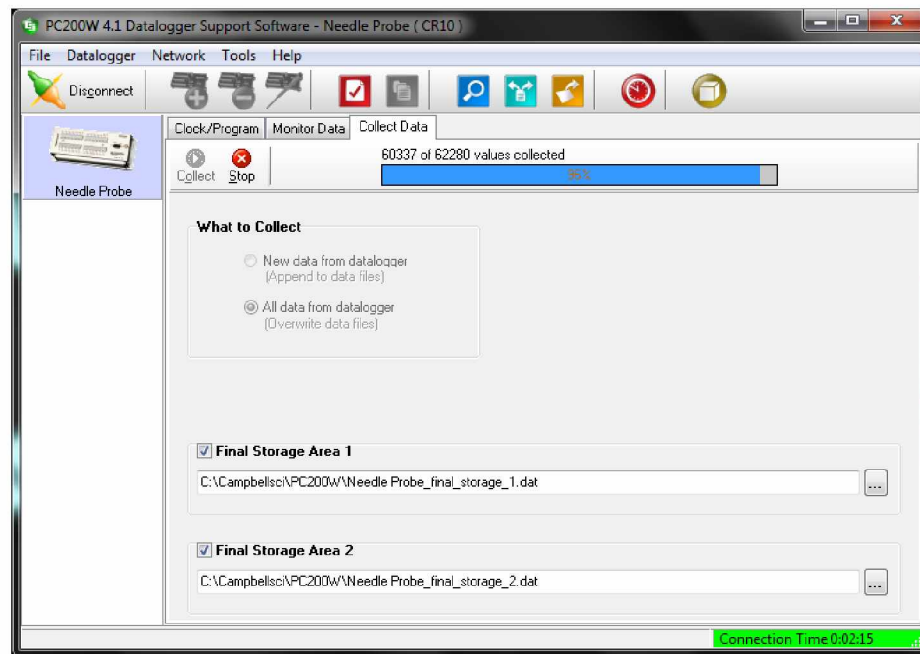


Figure 4.2: A screenshot of PC200W, the software used to pull data off the CR10X data logger.

tures that situating the needle without ruining the snowpack can be quite a challenge.

Along with each conductivity measurement, the height from the ground—measured with a tape measure—and the angle of insertion—measured with a protractor and a plumb bob—were recorded as metadata. In addition, for each series of measurements at a particular region, the density of the snow is also measured with a cardboard cylinder (used as a control volume) and a scale.

4.4 Benchtop Tests

A standard used to benchmark needle probes is to measure the conductivity of a large Nalgene bottle full of glycerine and surrounded in insulating foam to mitigate changes in room temperature. Glycerine was chosen because it has almost the same conductivity as water and does not readily convect. Moreover, glycerine does not leave an air gap between the needle and the surrounding medium like many porous materials, such as snow, will.

A method for testing anisotropic measurements by using alternating layers of more-

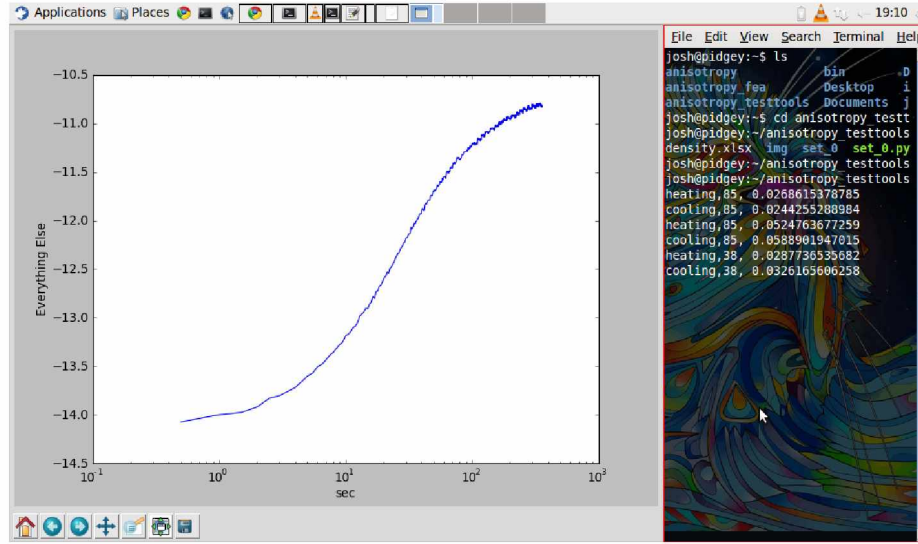


Figure 4.3: A plot of temperature vs. time from a real-world measurement of snow. Each curve must be analyzed by-hand to check for such effects as convection, as seen on the right-hand side of this curve.

conductive and less-conductive materials has been devised, based on this glycerine benchmark test. However, instead of glycerine, the materials used are table salt and table sugar.

4.5 Raw Materials for the Anisotropic Composite

Salt and sugar's conductivities alone were both measured using the needle probe apparatus. These measurements resulted in conductivities of 0.225 W/m/K and 0.106 W/m/K, respectively.

Assuming alternating layers of equal thickness, the anisotropic thermal conductivities in the aggregate should be:

$$k_{xy} = \frac{1}{2} (k_{\text{salt}} + k_{\text{sugar}}) = \boxed{0.166} \quad (4.1)$$

$$k_z = 2 \left(\frac{1}{k_{\text{salt}}} + \frac{1}{k_{\text{sugar}}} \right)^{-1} = \boxed{0.144} \quad (4.2)$$

$$\frac{k_z}{k_{xy}} = \boxed{0.870} \quad (4.3)$$



Figure 4.4: A close-up shot of tested snowpack.

Despite the conductivity of table salt being roughly twice that of sugar, the anisotropic conductivity ratio is fairly close to one, meaning that the anisotropy of the experimental medium, while significant, is relatively weak. Advantageously, however, salt and sugar are relatively inexpensive media to work with.

4.6 Apparatus for Containing Anisotropic Composite

In order to effectively change the directions of anisotropy, a foam box for the nalgene bottle was built that could be rotated on an axis and clamped in place. The resulting angle from the horizontal can be measured with a protractor. The apparatus was designed with gels such as glycerine in mind, such that the layers would self-level. However, because powders were used instead of gels, leveling had to be done by hand, usually with a spoon. The sugar was dyed green in order to differentiate it from the salt.

Table 4.1: Raw results of salt and sugar measurements after calculating conductivity. The multiple results were averaged for the purpose of predicting anisotropic conductivity of an alternately-layered medium.

Material	#	Heating	Cooling	Average	Standard Deviation
Pure salt	1	0.222	0.220	0.225	0.015
	2	0.218	0.256		
	3	0.216	0.219		
Pure sugar	1	0.108	0.113	0.106	0.008
	2	0.098	0.109		
	3	0.094	0.113		

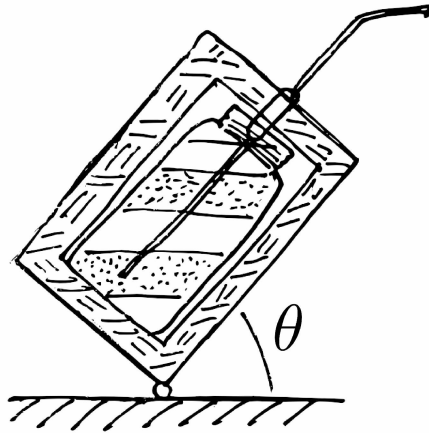


Figure 4.5: An illustration of the apparatus used in benchtop measurements. The apparatus was designed to tilt in order to cause alternating layers of self-leveling materials to meet the needle at a given angle. However, the materials actually used were not self-leveling, meaning the tilting apparatus was of limited utility.



Figure 4.6: A photograph of the benchtop measurement apparatus in use, at 30 degrees.

Chapter 5

Results and Interpretation

5.1 Parameters and Nondimensionalization

For most analyses, many of the parameters and results are non-dimensionalized. In particular, instead of separate parameters for k_z and k_{xy} , k_z and k_{eff} are both normalized by k_{xy} . Numerical experiments verified that this is permissible, as numerical experiments with different k_z and k_{xy} parameters but equivalent ratios k_z/k_{xy} resulted in very similar ratios of k_{eff}/k_{xy} .

Angle is an exception. In this analysis, all angles are given as degrees from the horizontal (xy) plane.

5.2 Numerical vs. Analytical Predictions

A 3-D plot of the numerical and analytical predictions may be seen in Figures 5.1. This plot shows that the two approaches to predicting measured conductivity as a function of angle and anisotropy ratio have similar trends. However, there are important disagreements which must be resolved.

Many details may be seen more readily in two-dimensional plots. In particular, Figure 5.2 shows theoretical predictions from both the analytical and numerical model as a function of anisotropic conductivity ratios, sliced by angle, while Figure 5.3 shows theoretical predictions from both methods as a function of angle, sliced by anisotropic conductivity ratios. Figure 5.2 readily shows that an increase in conductivity ratio above 1 has a weaker effect on effective conductivity for the numerical model than in the analytical model. This may be due to edge effects decreasing the effective thermal conductivity in the numerical model by conducting heat axially from the needle. The analytical model, in contrast, does not model edge effects, as it models an infinitely long needle like the original needle probe method. For the isotropic case, these edge effects have been studied and quantified analytically by other researchers. [Blackwell, 1956]

Figure 5.3 also shows the smoothing seen in Figure 5.2, but also readily shows the predictions of both models for the isotropic case ($k_{\text{eff}}/k_{xy} = 1$). Both models correctly predict that effective thermal conductivity is not a function of angle for the isotropic case. However, it is also clear that the numerical model over-predicts k_{eff} by at least 10%.

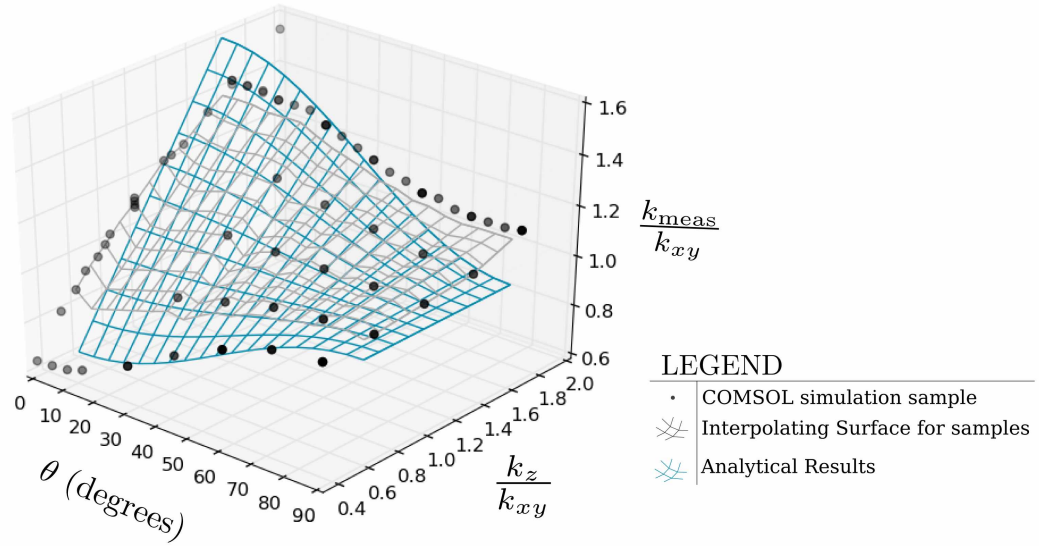


Figure 5.1: A comparison of the numerical results and the analytical theory shows general agreement. Grey dots represent numerical simulation results, the grey surface represents an interpolating surface of the dots, and the blue surface represents the analytical model. Disagreement between the two may be due to edge effects and/or numerical model convergence issues.

The dominant cause for this discrepancy is believed to be due to using a model with too coarse of a mesh. The convergence study results show that, while the time/temperature curves look largely the same (Figure 5.4), that the minor differences are magnified when taking the derivative with respect to $\ln(t)$ such that the coarse grid reports a thermal conductivity of about 110% of the finer grid (Table 5.1). This difference is of the same order of magnitude as the difference seen between the numerical predictions and the analytical predictions, and it is believed that running the same simulations at a finer grid would resolve most of the difference seen, particularly in the isotropic case.

However, some of this may also be due to edge effects, as evidenced by the cluster of points at the zero angle (More readily seen in Figure 5.5), where it can be seen that, in fact, the predictions for the numerical method are a very weak function of k_{xy} and not just the ratio of the anisotropic conductivities. Simply repeating the experiments at higher

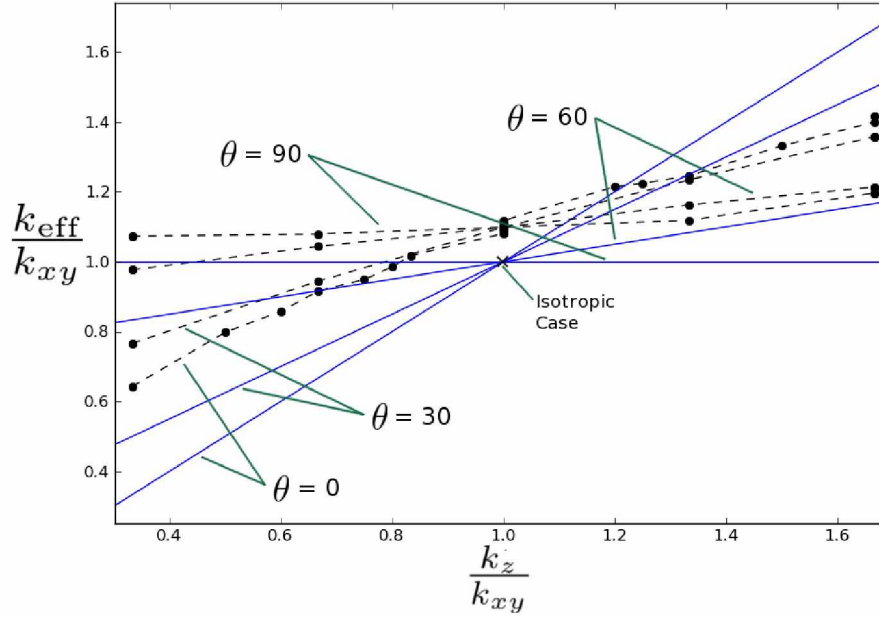


Figure 5.2: Slices of theoretical predictions by angle. Black points connected by dashed lines represent numerical results, while solid blue lines represent analytical theory. It can be seen that the analytical theory predicts measured conductivity to be a stronger function of angle than the numerical data at higher conductivity ratios.

resolutions of mesh should be able to resolve this problem.

Figure 5.5 shows predictions for the special case of $\theta = 0$, where the needle is oriented parallel to the planes of isotropy. This special case is of interest because previous needle probe measurements have determined effective conductivities at this angle, which may or may not be accurate representations of k_z , the vertical thermal conductivity, which is what is measured by a guarded hot plate apparatus and is the conductivity of interest to climatologists modeling heat transfer between the atmosphere and soil. If predictions for k_{eff} are *equal* to k_z , then the measured k_{eff} accurately reflects k_z . This agreement between measurements seems unlikely for anisotropic measurements, and in fact the numerical model shows the expected trend of $k_{\text{eff}} > k_z$ for low conductivity ratios, and $k_{\text{eff}} < k_z$ for higher conductivity ratios. However, the analytical model predicts that $k_{\text{eff}} = k_z$ for all anisotropy ratios. While the analytical model shows expected behavior for the isotropic case and shows the general trends one would expect, this surprising result casts doubt

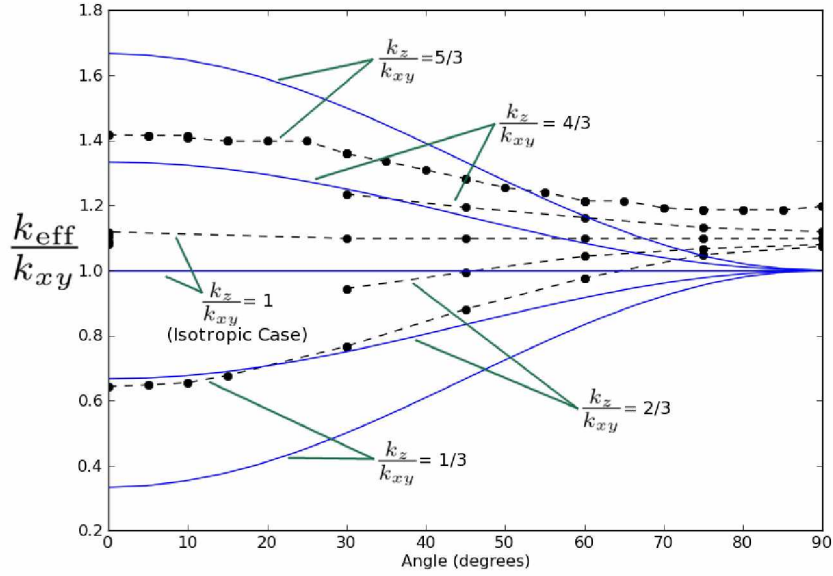


Figure 5.3: Slices of theoretical predictions by k_z/k_{xy} . Black points connected by dashed lines represent numerical results, while solid blue lines represent analytical theory. It can be seen that the analytical theory is perfect for the isotropic case ($k_z/k_{xy} = 1$), while the numerical experiments report larger-than-expected values.

onto the validity of the analytical model.

5.3 Benchtop Measurements

It may be seen that there is a significant amount of variation between benchtop measurements using the needle probe method in Figure 5.6 and Table 5.3, even accounting for obviously failed measurements such as the removed outlier in Table 5.2. This is likely due in part to the nature of numerical derivatives as well as the relatively unpredictable behavior of porous materials. Given this variation, it is difficult to see which of the two models (analytical or numerical) is more appropriate.

Based on a general curve fit, the benchtop results show a slight upward slope (Figure 5.6) as expected. However, given the variation in the benchtop results, it is statistically possible that angle has absolutely no effect on the reading. This could be fixed with more

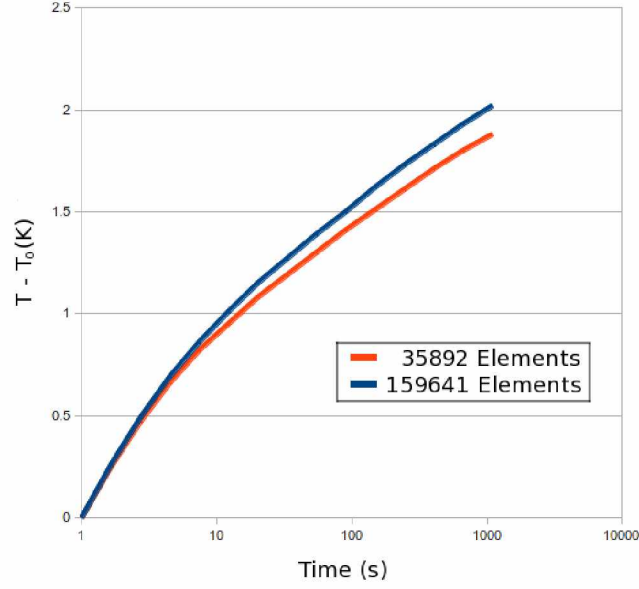


Figure 5.4: A comparison of two $T(t)$ curves from equivalent simulations with different fineness of mesh. These two curves appear quite similar, but their long- time slopes are measurably different

careful, exacting standards in the construction of the anisotropic materials, more measurements at each angle, and measurements at more angles. In other words, given the variance of the measurements, many more measurements would have to be made in order to reach any statistically significant conclusions, at least given the relatively low amount of anisotropy of the sample.

Also given this variation and the relatively weak levels of anisotropy in the sample, even with more measurements it could still prove difficult to deduce the degree of anisotropy of the sample with this data and a curve fit to either the numerical or analytical predictions alone.

5.4 In-Situ Snow Measurements

Due to the difficulty in taking snow measurements, very few snow measurements were successfully completed (Table 5.4). This, on top of the inherent variation between measurements seen in the method, snow measurements are also inconclusive. However, the mea-

Table 5.1: A comparison of k_{eff} from two equivalent simulations with different fineness of mesh. Despite the similarities in time/temperature curves, the resulting conductivity calculations differ by nearly 10 %. Units are in W/m·K.

	Slope
35892 elements	0.215
159641 elements	0.198
% Error	8.64 %

measurements taken *do* indicate anisotropy to a greater degree of confidence than the benchtop measurements, as may be seen in Figures 5.8 and 5.9. Like the case of the benchtop measurements, with so few measurements a curve fit against either method of prediction is unlikely to yield useful results.

While the degree of anisotropy of the snow sample is unknown, it is known that the measurements indicate a k_z/k_x of less than one, which is indicative of the aggregate anisotropy seen from a geometry of alternating layers, and not of the particular structural anisotropy that is seen in some types of snow such as depth hoar *or* the sort of anisotropy that could explain the discrepancies between needle probe and guarded hot plate measurements. This is not surprising, as the measurements were taken in a layered region in order to increase the chances of detecting anisotropy.

To put the snow measurements in context for comparison in future studies, Table 5.5 shows the snow density in the region of snow that these measurements were taken.

5.5 Ramifications

These results indicate that anisotropy is possible to measure in snow. However, due to variance between measurements, this may be more difficult than hoped for. While only two perfect measurements would likely be required to ascertain the anisotropic thermal conductivity of snow given a proven model, the variance seen in these measurements suggests that many more measurements would be required—somewhere on the order of five to ten measurements per angle would likely be required for statistically significant results.

However, horizontal measurements do reflect vertical conductivity to a degree, and for

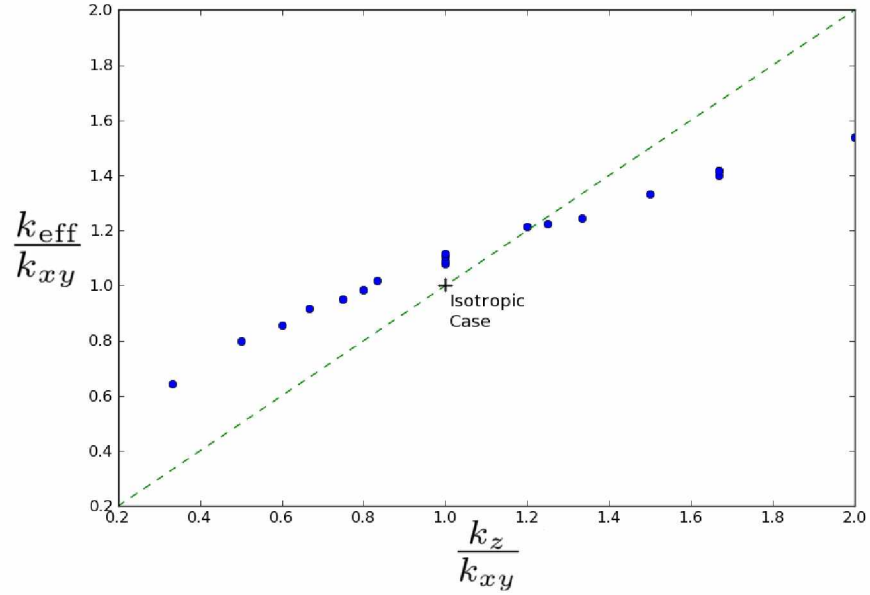


Figure 5.5: Theoretical predictions for the special case of $\theta = 0$, when the needle is oriented horizontally. Blue points represent numerical solution, while green shows the line where $k_{\text{eff}} = k_z$, where measured conductivity and vertical conductivity are the same. The analytical predictions are indistinguishable from this line, and are therefore not plotted.

small levels of anisotropy (likely if the anisotropy is on the aggregate level only) horizontal measurements may be sufficient for ascertaining vertical conductivity.

Given that anisotropy was detected in the snowpack, it is possible that snow anisotropy is responsible for the discrepancies between guarded hot plate measurements and needle probe measurements. However, because aggregate anisotropy would lead us to predict higher in-plane conductivity than out-of-plane conductivity, only structural anisotropy can explain this discrepancy.

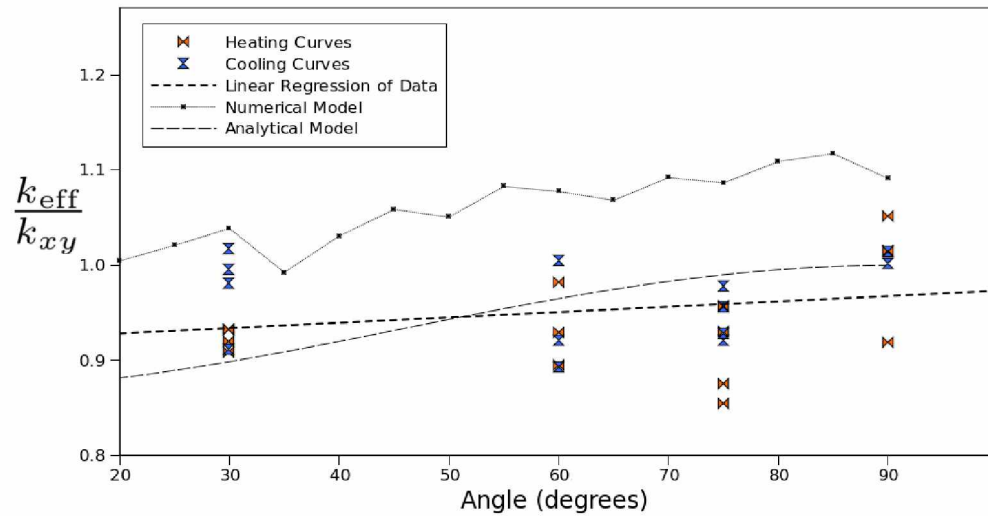


Figure 5.6: A comparison of the benchtop measurements with the numerical and analytical predictions for alternating layers of salt and sugar, given the calculated anisotropic thermal conductivity.

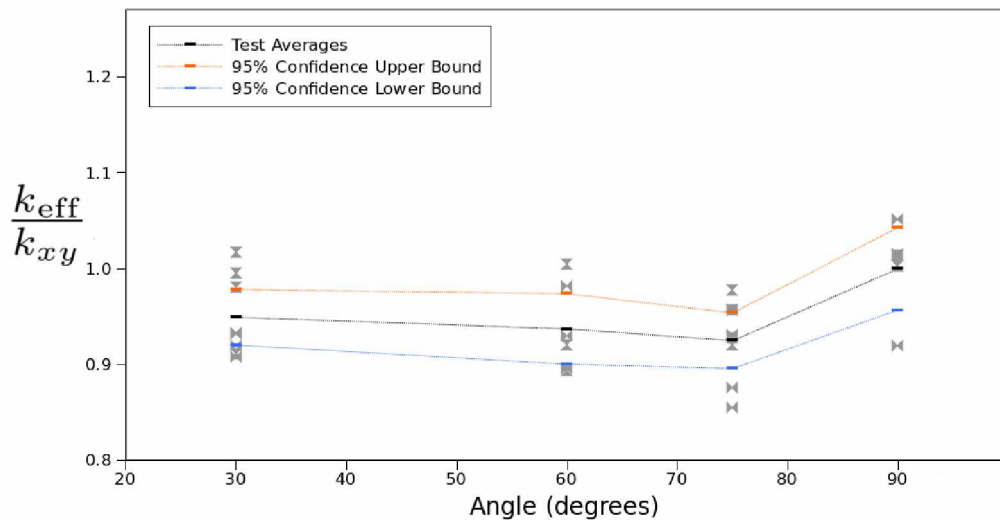


Figure 5.7: Upper and lower bounds of 95 % confidence in thermal conductivity measurements of the salt and sugar layered samples. This analysis indicates that the measurements can not statistically exclude a null hypothesis.

Table 5.2: Raw data from the benchtop measurements. Note that one of the cooling curve measurements is striked out. This is because, when examined, it is clearly an outlier. Units are in $\text{W/m}\cdot\text{K}$.

Angle (degrees)	#	Heating	Cooling
90	1	0.223	0.243
	2	0.247	0.246
	3	0.256	0.318
75	1	0.213	0.224
	2	0.233	0.232
	3	0.208	0.226
	4	0.226	0.238
60	2	0.239	0.244
	3	0.226	0.224
	4	0.218	0.217
30	1	0.227	0.238
	2	0.226	0.247
	3	0.221	0.242
	4	0.223	0.221

Table 5.3: Basic statistics on normalized benchtop measurements. Units are in $\text{W/m}\cdot\text{K}$.

Angle	k_{meas}/k_{xy}^-		
	Mean	Standard Deviation	95% Confidence
90	1.000	0.0491	0.0431
75	0.923	0.0419	0.0291
60	0.937	0.0459	0.0367
30	0.949	0.0420	0.0291

Table 5.4: Conductivity results from the snow measurements. Units are in W/m·K.

Angle	Heating	Cooling
0	0.0289	0.0321
5	0.0269	0.0244
45	0.0290	0.0327
52	0.0288	0.0326

Table 5.5: Measured and derived measurements for snow density. Units are in W/m·K.

Control Volume	736.76 mL
Mass	161.14 g
Density	0.219 g/mL
	219 kg/m ³

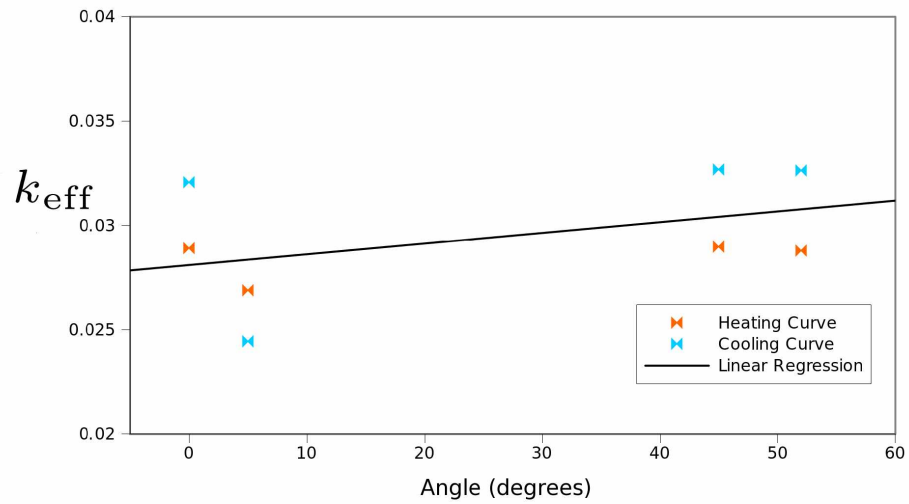


Figure 5.8: Conductivity measurements in roughly the same layer of snow at various angles. Even a limited amount of in-situ snow measurements suggest a degree of anisotropy.

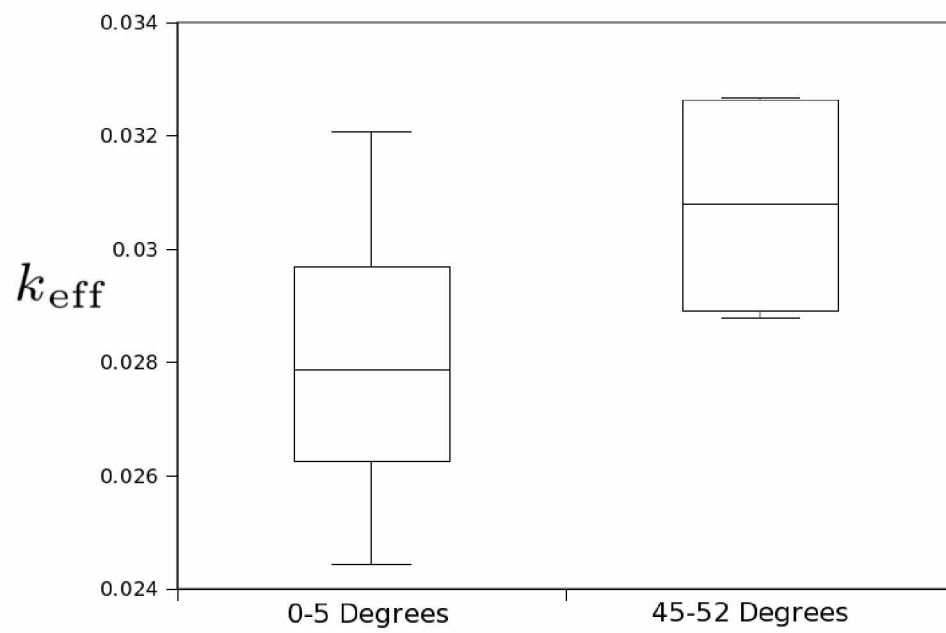


Figure 5.9: These boxplots give a general idea of the differences in measurements between the near-horizontal angle and the more oblique ones in snow.

Chapter 6

Future Work

6.1 Introduction

This is clearly not the end of research on this method. While the groundwork has been laid, there are plenty of avenues which need more study.

6.2 Assumptions in the Analytical Approach

In the analytical model presented here, the needle is assumed to be of zero thickness, and yet an average temperature over an ellipse is taken which represents the needle surface. The model seems to work relatively well; however, there are other ways to represent such a problem that may yield more accurate results. For example, one may be able to approach the problem in terms of a needle with a finite thickness, in which case the solution to the problem should be an infinite series of Bessel functions in the isotropic case. One may be able to tackle the problem from a finite-thickness needle approach for better results. In addition, a refined model could account for edge effects. [Blackwell, 1956]

Another recommendation for future study is to build a hybrid method, which uses the projection technique developed for the analytical method to specify a 2D problem, but uses a finite element method to solve the 2D problem. Since this new finite element model would not account for edge effects, it could be used to help the causes of the discrepancies between the models. Moreover, since 2D problems are much easier to solve, experiments may be ran at very high mesh resolutions as compared to the 3D problem.

6.3 Extended Convergence Study

While a convergence study for the numerical model was undertaken, it was relatively informal, and executed for only one particular configuration of parameters. A more thorough investigation of the convergence properties of the numerical model should likely be undertaken, especially in light of the 10% error observed in the convergence study done in this work.

6.4 Improved Benchtop Method

The benchtop apparatus was based on the use of glycerine and agar gels for calibration. However, the current anisotropic benchtop method uses sugar and salt. This makes the proper layering of the materials difficult. Moreover, the device is limited to a tilt factor of 30 degrees from horizontal due to the location of the bottle's opening. Presumably, this method could be improved upon to allow for more accurate material layering and for an increased range of needle angles.

In addition, the materials used only lend themselves to an anisotropy ratio of 87%. There is plenty of room for improvement, in terms of suitable materials. Figure 6.1 shows what material conductivity ratios are required to achieve a given anisotropic conductivity ratio, assuming equal-thickness layers. While salt and sugar fare poorly, real-world materials have a wide range of conductivities such that two materials with sufficiently different conductivities should be possible to find.

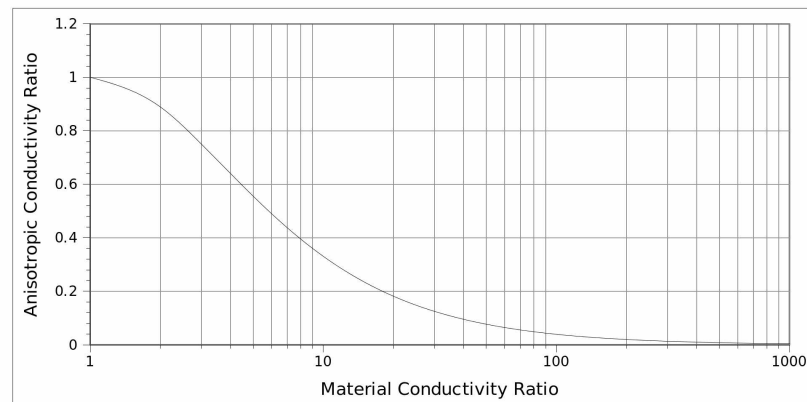


Figure 6.1: Anisotropic Conductivity Ratio vs. Material Conductivity Ratio for the layered geometry used in the benchtop experiments. It may take a very large material conductivity ratio in order to achieve a relatively minor anisotropic conductivity ratio.

6.5 Comprehensive Benchtop Measurements

While enough benchtop measurements were taken to give a vague idea as to the effectiveness of this measurement technique, there were not nearly enough measurements to give a statistically valid conclusion regarding the slight trend we were looking for. In addition

to improving the benchtop method, there need to be many more measurements taken, in order to draw statistically valid conclusions.

6.6 Comprehensive In-Situ Measurements

The in-situ measurements presented in this document are very limited in scope. One could easily spend much more time taking more measurements on more snow at more angles, in order to better quantify the degrees of anisotropy in natural snowpacks. More measurements would allow for a rigorous statistical analysis that can show a trend with high confidence.

6.7 Exploration of the Cooling Curve

In the numerical and analytical models, the cooling curve is all but ignored. It is believed that cooling curve models will yield analogous results, but this has not been tested. Because of the importance of cooling curve measurements in the real world (as they effectively double the number of measurements in a sample and act as a consistency check on heating curve results), verification of these expectations of analogous behavior should occur.

6.8 A Method for Determining Anisotropic Thermal Conductivity From Measurements

While this document lays the groundwork for determining anisotropic thermal conductivity from measurements, a reliable method for converting measurements into anisotropic thermal conductivities has not been found. This is, in part, due to the discrepancy between the numerical and analytical approaches to predicting effective thermal conductivity measurements as a function of angle and degree of anisotropy, since the accuracy of the models is unknown. Moreover, the lack of solid empirical data means that there is no evidence to support either theory, outside of the isotropic case.

Given a reliable theory and data, one approach for ascertaining thermal conductivity would be to find the degree of anisotropy for which the data as a function of angle best fits the predicted k_{eff} curves as a function of k_{xy} and k_z .

One possibility is that, instead of focusing on the measured conductivities as a function of angle, that one should focus on the *change* in measured conductivities with respect to

angle. This is because, while the actual predicted values for k_{eff} disagree, the predicted percent difference between two measurements may be less so, particularly for instances of smaller anisotropy ratios. In fact, given low enough degrees of anisotropy, it may be sufficient to compare linearizations of k_{eff} vs. θ curves. Only with more research can this conjecture be shown to be valid.

One suggestion for collecting data to determine the correct theory is to take measurements at the most extreme angles possible. In the case of measuring conductivity at the center of the snowpack, 45 degrees from horizontal is about the practical limit. However, tests conducted at the top of the snowpack would be able to record measurements at 90 degrees from horizontal, the angle at which the other expected extreme value is expected to occur. Given this data, determining the correct theory may be easier. The author recommends, in addition to other measurements, that this experiment is conducted in future research.

Chapter 7

Conclusions

An analytical model based on the isotropic solution outlined by Carslaw and Jaeger has been modified in order to predict thermal conductivity measurements of anisotropic materials as a function of insertion angle. This model uses a linear transformation to pose the problem in a form equivalent to the isotropic problem. However, in order to get meaningful results, the model requires calculating the average temperature along an ellipse.

A 3-D finite element numerical model has also been built in order to predict thermal conductivity measurements of anisotropic materials as a function of insertion angle. While the base model is simple by finite element model standards, the number of parameters iterated through is somewhat unusual for finite element modeling, and taxes the abilities of the software used.

According to both numerical and analytical theories, anisotropic thermal conductivity should cause predictable changes in needle probe heating curve measurements as a function of angle. While the two models show similar trends, there are significant differences between the two predictions. These may in part be explained by the 3D model's handling of edge effects, and in part due to the numerical model having too coarse of a mesh to accurately model the problem.

Measurements of engineered anisotropic materials are promising, but far from complete. A basic, repeatable method has been designed and tested. The initial results indicate the expected trend; however, due to the relatively low amount of anisotropy in the material, the variance in measurements of the thermal conductivity measurement method and a disappointingly low amount of measurements, a null hypothesis is impossible to rule out.

Similarly, in-situ measurements of snow have been made, but due to difficulties in snow measurement very few useful data points were collected. However, the results do indicate anisotropy in the snow. This detected anisotropy is likely due to aggregate effects of layering, and not due to the structural anisotropy that could potentially explain the differences between guarded hot plate and needle probe measurements.

Based on these results, anisotropy is likely detectable in snow and possible to determine, though it will require more measurements than perhaps hoped for. If this is

anisotropy is to explain the discrepancies between guarded hot plate and needle probe measurements, it must be shown that structural anisotropy in snow is significant enough not only to explain the guarded hot plate/needle probe discrepancies itself, but also enough to counteract any anisotropy being caused by layering in snow. This research suggests that structural anisotropy *may* be able to explain discrepancies between guarded hot plate measurements and needle probe measurements, though, with a lack of hard data, the answer to this problem is still unknown. More experiments will be required to know for sure.

Inquiry on this method is far from complete. First, there are significant discrepancies between the analytical and numerical theories which must be resolved. Second, there are not enough measurements to benchmark either method. Because of these issues, a suitable, robust method for ascertaining anisotropic thermal conductivity from multiple needle probe measurements has not been sufficiently developed. Development of such a method will be instrumental not only in benchmarking the theoretical predictions but also critical in answering the question of what causes the disagreement between guarded hot plate and needle probe measurements.

Bibliography

- Blackwell, J. (1956), The axial-flow error in the thermal-conductivity probe, *Canadian Journal of Physics*, 34, 412–417.
- Carslaw, H., and J. Jaeger (1959), *Conduction of Heat in Solids*, 2nd ed., Oxford University Press.
- Ewen, J., and H. Thomas (1987), The thermal probe—a new method and its use on an unsaturated sand, *Geotechnique*, 37(1), 91–105.
- Lunardini, V. J. (1981), *Heat Transfer in Cold Climates*, Van Nostrand Reinhold Company.
- Pitman, D., and B. Zuckerman (1968), Effective thermal conductivity of snow at -88°C , -27°C and -5°C , *Smithsonian Astrophysical Observatory Special Report*, (267).
- Pratt, A. (1969), Heat transmission in low conductivity materials, *Thermal Conductivity*, 1, 301–402.
- Sturm, M., and J. Johnson (1991), Natural convection in the subarctic snow cover, *Journal of Geophysical Research*, 96(B7), 675–671.
- Sturm, M., and J. B. Johnson (1992), Thermal conductivity measurements of depth hoar, *Journal of Geophysical Research*, 97(B2), 2129–2139.
- Sturm, M., D. K. Perovich, and J. Holmgren (2002), Thermal conductivity and heat transfer through the snow on the ice of the beaufort sea, *Journal of Geophysical Research*, 107(C21).

Appendix A

Code Used for Chapter 2

A.1 model.py

```
import json

from math import pi
from numpy import log, sin, cos, sqrt, array, \
    arange, hstack, linspace, dot
from functools import reduce

def elliptical(fxn, ecc):
    from scipy.integrate import quad
    from math import pi
    from numpy import sin, cos, sqrt
    from types import FunctionType, BuiltinFunctionType

    #API trickery.
    if type(fxn) != FunctionType and type(fxn) != BuiltinFunctionType:
        fxn2 = lambda ecc, th: fxn
    else:
        fxn2 = fxn

    #The heavy lifting.
    return quad(lambda th: fxn2(ecc,th) *
                sqrt(cos(th)**2.0 + (ecc*sin(th))**2.0 ),
                0, 2*pi)[0]

def Tavq(k_x, k_y, q, t):
    """
    given scalar r0, k_x, k_y and 1-d time, this returns a curve with
    the same slope at Tavq(t) for long T. May refactor.
    """
    return (4*pi*k_x/q)*array([elliptical(log(time) , k_x/k_y)/
                               elliptical(1, k_x/k_y) for time in t])

def kmeas(k_x, k_y, q, t):
```

```

from numpy import polyfit
"""
Does a quick linear curve fit
"""
return (q/4/pi)*polyfit(log(t), Tavg(k_x, k_y, q, t), 1)[0]

def rot(th, axis):
    from numpy.linalg import norm
    from numpy import sin, cos, eye, outer, cross
    if axis == "x":
        axis = array([1,0,0])
    elif axis == "y":
        axis = array([0,1,0])
    elif axis == "z":
        axis = array([0,0,1])
    else:
        axis = axis/norm(axis)
    oh = outer(axis, axis)
    return oh + cos(th)*(eye(3) - oh) + sin(th)*cross(axis, eye(3))

def proj(matrix, rot):
    from numpy import eye, hstack, vstack, newaxis
    from numpy.linalg import eig
    return tuple(
        eig(
            reduce(dot, [ vstack((
                hstack((
                    eye(2),
                    array([0, 0])[ :, newaxis] )),
                array([0, 0, 0]))),
                rot,
                matrix,
                rot.T ])))[0])[0:2]

```

```

if __name__=="__main__":
    from numpy import diag, logspace, meshgrid
    from math import pi
    from progressbar import ProgressBar

    angles = range(0, 91) # Lots angles :D

    ks = arange(0.2, 0.4, 0.05) # Some ks
    (k_xy, k_z) = meshgrid(ks, ks)
    k_xy = k_xy.flatten()
    k_z = k_z.flatten()

    q = 0.5 #Like in sims
    t = hstack(( logspace(0.1,1.0,15),
                 logspace(1.0,3.0,15) ))

    results = []
    progress = ProgressBar()
    for th in progress(angles):
        for i in xrange(k_xy.shape[0]):
            (k_xp, k_yp) = proj( diag([k_xy[i], k_xy[i], k_z[i]]),
                                rot(pi/180*(90-th), 'y'))
            results.append([ th,
                             k_z[i]/k_xy[i],
                             kmeas(k_xp, k_yp, q, t)/k_xy[i]])
    for row in results:
        print(', '.join(map(str, row)))

```

Appendix B

Code Used for Chapter 3

B.1 worker.m

```
%worker.m
%does the working

function worker(kxy,kz)
    load('angles.mat', 'angles');
    angles
    [kxy,kz] = meshgrid(kxy,kz);

    % The commented-out 'flreport' gives one the option of suppressing
    % graphical output from COMSOL. This is useful if one wants to run
    % COMSOL in a headless manner. Unfortunately, COMSOL 3.5a on the ARSC
    % systems has extreme difficulties running in headless batch mode.
    %flreport('off');

    params=struct('rsnow', 0.4, ...
                  'rneedle', 0.00025, ...
                  'lneedle', 0.1, ...
                  'density_snow', 200, ...
                  'density_needle', 8000, ...
                  'cp_snow', 2050, ...
                  'cp_needle', 460, ...
                  'q_needle', 0.5, ...
                  'k_needle', 160, ...
                  'time', [logspace(0.1,1,15) logspace(1,3,15)], ...
                  'angles', angles );

    saveroot=['./solutions-' date '/'];

    mesh = mesher(0,params);
    for angle=angles,
        try
            solutions = arrayfun(@(x,y) solver(x,y,mesh,angle,params), ...
```

```

        kxy,kz, 'UniformOutput', false);
    save solutions
    fprintf('Fitting solutions...\n');
    solutions = {cellfun(@(tsd) {fitter(tsd{1}(1,:),tsd{1}(2,:), ...
        0.999,params), ...
            tsd{1}, tsd{2}}), ...
        solutions, 'UniformOutput', false)};
    fprintf('A solution set just completed.');
```

catch exception

```

        system([ 'echo "Exception occurred on" `hostname` ' ...
            '| mutt -s "Exception occurred--' exception.message ...
            '" josh.holbrook@gmail.com' ]]);

    end

    angles = angles(2:length(angles));
    save('angles.mat', 'angles');
    %solutions
    mkdir(saveroot);
    save([ saveroot 'solution-' num2str(angle) ], ...
        'solutions','angle','kxy','kz','params');

    end

    % Emails me when everything's done
    system([ 'echo "Results completed on " `hostname` ' ...
        '| mutt -s "Results Completed" ' ...
        '| josh.holbrook@gmail.com' ]]);
    system('touch down');
```

end

B.2 mesher.m

```

% COMSOL Multiphysics Model M-file
% Generated in part by COMSOL 3.5a
% (COMSOL 3.5.0.608, $Date: 2009/05/11 07:38:49 $)
```



```

% the rest of it modified by Joshua Holbrook

function fem=mesher(angle,params)
    % mesh_generate(angle)
    % generates a mesh for the given angle.

    fprintf(['meshing for angle=' num2str(angle) '...\n']);
    flclear fem

    % COMSOL version
    clear vrsn
    vrsn.name = 'COMSOL 3.5';
    vrsn.ext = 'a';
    vrsn.major = 0;
    vrsn.build = 608;
    vrsn.rcs = '$Name: v35ap $';
    vrsn.date = '$Date: 2009/05/11 07:38:49 $';
    fem.version = vrsn;

    % Geometry
    g1=sphere3(num2str(params.rsnow), ...
        'pos',{'0','0','0'}, ...
        'axis',{'0','0','1'}, ...
        'rot','0');
    g2=cylinder3(num2str(params.rneedle), ...
        num2str(params.lneedle), ...
        'pos',{num2str(-params.lneedle/2),'0','0'}, ...
        'axis',{'1','0','0'}, ...
        'rot','0');
    parr={point3(0,0,0)};
    g3=geomcoerce('point',parr);
    parr={point3(params.rsnow,0,0)};
    g4=geomcoerce('point',parr);
    parr={point3(0,params.rsnow,0)};
    g5=geomcoerce('point',parr);

```

```

parr={point3(0,0,params.rsnow)};
g6=geomcoerce('point',parr);
parr={point3(-params.rsnow,0,0)};
g7=geomcoerce('point',parr);
parr={point3(0,-params.rsnow,0)};
g8=geomcoerce('point',parr);
parr={point3(0,0,-params.rsnow)};
g9=geomcoerce('point',parr);

% Analyzed Geometry (?)
clear p s
p.objs={g3,g4,g5,g6,g7,g8,g9};
p.name={'ORIGIN','PT1','PT2','PT3','PT4','PT5','PT6'};
p.tags={'g3','g4','g5','g6','g7','g8','g9'};

s.objs={g1,g2};
s.name={'SNOW','NEEDLE'};
s.tags={'g1','g2'};

fem.draw=struct('p',p,'s',s);
fem.geom=geomcsg(fem);

% ODE Settings
clear ode
clear units;
units.basesystem = 'SI';
ode.units = units;
fem.ode=ode;

% Initialize mesh
fem.mesh=meshinit(fem, ...
    'hauto',5, ...
    'hgradsub',[2,1.1], ...

```

```

        'hmaxsub',[2,0.0005]);

% Refine mesh
fem.mesh=meshrefine(fem, ...
    'mcase',0, ...
    'rmethod','longest');

fem=multiphysics(fem);
end

```

B.3 solver.m

```

% COMSOL Multiphysics Model M-file
% Generated by COMSOL 3.5a
% (COMSOL 3.5.0.608, $Date: 2009/05/11 07:38:49 $)

function answer=solver(kxy,kz,fem,theta,params)
% solver(kxy,kz,mesh,params)
% uses comsol to pump out a solution using a given mesh-mat
% and a k-matrix in comsol format.

fprintf(['solving for kxy=' num2str(kxy) ...
    ' and kz=' num2str(kz) '...\n']);
% Application mode 1
clear appl
appl.mode.class = 'GeneralHeat';
appl.module = 'HT';
appl.shape = {'shlag(1,''J'')','shlag(2,''T'')'};
appl.sshape = 2;
appl.assignsuffix = '_htgh';
clear bnd
bnd.type = {'q0','cont'};
bnd.shape = 1;
bnd.ind = [1,1,1,1,2,2,2,2,2,1,1,1,1,2];
appl.bnd = bnd;
clear equ

```

```

equ.sdtype = 'gls';
% densities
equ.rho = {params.density_snow,params.density_needle};
equ.init = 0;
equ.shape = 2;
% Heat capacities
equ.C = {params.cp_snow,params.cp_needle};
% Wattage
equ.Q = {0,params.q_needle/pi/(params.rneedle)^2};
% Heat conductivities
arr = [cos(theta*pi/180), 0, sin(theta*pi/180);
       0, 1, 0;
       -sin(theta*pi/180), 0, cos(theta*pi/180)]; %rotation matrix
equ.k = {symmetric_tocell( ...
          arr*diag([kxy,kxy,kz])*(arr')), ...
          params.k_needle};
equ.ind = [1,2];
appl.equ = equ;
fem.appl{1} = appl;
fem.frame = {'ref'};
fem.border = 1;
fem.outform = 'general';
clear units;
units.basesystem = 'SI';
fem.units = units;

% Coupling variable elements
clear elemcpl
% Integration coupling variables
clear elem
elem.elem = 'elcplscalar';
elem.g = {'1'};
src = cell(1,1);
clear bnd
bnd.expr = {'T', {}, {'1', {}}};

```

```

bnd.ipoints = {{ '4', {} }, { '4', {} }};
bnd.frame = {{ 'ref', {} }, { 'ref', {} }};
bnd.ind = {{ '1', '2', '3', '4', '10', '11', '12', '13', ...
            '5', '6', '7', '8', '9', '14' }};
src{1} = {{}, {}, bnd, {} };
elem.src = src;
geomdim = cell(1,1);
geomdim{1} = {};
elem.geomdim = geomdim;
elem.var = { 'int_T', 'area' };
elem.global = { '1', '2' };
elemcpl{1} = elem;
fem.elemcpl = elemcpl;

% ODE Settings
clear ode
clear units;
units.basesystem = 'SI';
ode.units = units;
fem.ode=ode;

% Multiphysics
fem=multiphysics(fem);

% Generate GMG mesh cases
fem=meshcaseadd(fem, 'mgauto', 'anyshape');

% Extend mesh
fem.xmesh=mesnextend(fem);

% Solve problem
fem.sol=femtime(fem, ...
                'solcomp', { 'T' }, ...
                'outcomp', { 'T' }, ...
                'blocksize', 'auto', ...

```

```

        'tlist', params.time, ...
        'estrat',1, ...
        'tout','tlist', ...
        'linsolver','gmres', ...
        'itrestart',100, ...
        'prefuntype','right', ...
        'prefun','gmg', ...
        'prepar',{'presmooth','ssor','presmoothpar',{'iter',3,'relax',0.
        'stopcond','0.06-int_T/area', ...
        'mcase',[0 1]);

% Save current fem structure for restart purposes
fem0=fem;

% Plot solution
%{
postplot(fem, ...
        'slicedata',{'T','cont','internal','unit','K'}, ...
        'slicexspacing',5, ...
        'sliceyspacing',0, ...
        'slicezspacing',0, ...
        'slicemap','Rainbow', ...
        'solnum','end', ...
        'title','Time=100    Slice: Temperature [K]', ...
        'grid','on', ...
        'campos',[-2.636014311828346, ...
                  -3.4353207343472505, ...
                  2.4999999999999996], ...
        'camtarget',[0,0,0], ...
        'camup',[0,0,1], ...
        'camva',41.213465344831754);
%}

% Integrate
T_thermistor=postint(fem,'T', ...

```

```

        'unit','K', ...
        'recover','off', ...
        'dl',8, ...
        'edim',0, ...
        'solnum','all');

% Integrate
T_surf_avg=postint(fem,'T', ...
        'unit','', ...
        'recover','off', ...
        'dl',[1,2,3,4,10,11,12,13], ...
        'edim',2, ...
        'solnum','end');

answer=[fem.sol.tlist; T_thermistor,T_surf_avg];
angles = params.angles(2:length(params.angles));

%flsave(['fem-' num2str(theta) '-' num2str(kxy) ...
%       '-' num2str(kz) '.mph']);

save('angles.m', 'angles');
end

```

B.4 fitter.m

```

function k = fitter(t,T,rset,params)
    logt = log(t(t>1));
    T = T(t>1);

    disp('Finding linear portion...');
    for i=1:length(logt)-1
        C = corrcoef(logt(i:length(logt)), T(i:length(T)));
        r = sqrt(C(2,1));
        if r > rset %adjust this to get 'good' values
            disp(['linear fitting to ' num2str((length(logt)-i)) ...
                ' points from t=' num2str(exp(logt(i))) ...

```

```

        ' to t=' num2str(exp(logt(length(logt)))) '...'];
    x = polyfit(logt(i:length(logt)),T(i:length(T)), 1);
    break
end
end

%plot(logt,T,'*');
%hold on;
%plot(logt, x(1)*logt + x(2));
k = (params.q_needle)/(4*pi*x(1));

end

```

B.5 assembler.m

```

function answers=assembler(directory)
    cd(directory);
    d = dir();
    answers = [];
    for i=3:length(d),
        disp(['Opening ' d(i).name '...']);
        load(d(i).name);
        answers = [answers, solutions];
    end
    cd('..');
end

```

B.6 reFitter.m

```

function fixed=reFitter(broked, r, params)
    fixed = broked;
    for i=1:length(fixed),
        fixed{i} = cellfun(@(kset) { ...
            fitHelper(kset{2}, r, params), ...
            kset{2}, kset{3}}, fixed{i}, 'UniformOutput', false);
    end
end

```



```

function fitted=fitHelper(tT, r, params)
    fitted = fitter(tT(1,:),tT(2,:),r, params);
end

```

B.7 analyzer.m

```

%analyzer
%Does some analyzing of the simulation results
%breaks for [kxy,kz] != meshgrid(ks,ks)

%Solutions location
%load solutions-19-Sep-2010/solutions-all.mat;

%Things I already know :)
%ks = linspace(0.2, 0.4, 6);
%ks = [0.2,0.4];
%[kxy, kz] = meshgrid(ks, ks);
%[kzy, kz] = meshgrid(0.3, 0.5);
%ks = 1;
%angles = 0:15:90;
%angles = 0:5:90;
%angles = [0 90];

%For an obvious color gradient, from red to blue right now.
colores = @(i,n) [sin((i/n)*pi/2), 0, cos((i/n)*pi/2)];

disp(['Showing overlaid plots (YES ALL OF THEM)' ...
    ' to make sure they "look" right:']);
figure;
hold on;
for theta = 1:length(angles)
    for i=1:length(ks)^2

        tT = answers{theta}{i}{2};
    end
end

```

```

        plot(log(tT(1,tT(1,:) > 1)),tT(2,tT(1,:) > 1), ...
            'color', colores(i,length(ks)^2));
    end
end

disp('Sanity checking results for isotropic cases');
figure;
hold on;
kmsold = 0 * cellfun(@(prison) prison{1}, answers{1});
for i=1:length(angles)
    %Extracts all the measured k's from the data
    % "prison" refers to cell representing particular
    % k combination in answers{theta}
    kms = cellfun(@(prison) prison{1}, answers{i});
    if kms == kmsold,
        disp('wtf exactly equivalent kms''s');
    end
    %diag(kms)
    %diag(kxy)
    errs = 100*(diag(kms) - diag(kxy))./diag(kxy);
    %Not necessary to be 3d anymore :)
    plot3(diag(kxy), errs, angles(i)*ones(size(diag(kxy))), ...
        '*-', 'color', colores(i,length(angles)) );
    xlabel('k_{actual}');
    ylabel('error (%)');
    zlabel('angle (degrees)');
end

disp('Figuring out T_surf_avg at time T:');
%figure;
%hold on;
for theta=1:length(angles)
    tavgs = cellfun(@(prison) prison{3}, answers{theta});
    try
        assert(all(all(tavgs< 0.001)));
    end
end

```

```

catch
    disp(['Warning: average surface temps are a bit high' ...
        ' at theta=' num2str(angles(theta))] );
    disp(tavgs);
end
if theta == length(angles)
    figure;
    hold on;
    contourf(kxy,kz,tavgs);
    colorbar;
    colormap('pink');
    title(['Average Surface Temperature at End of '...
        'Heating Curve Simulation for a representative angle']);
    xlabel('K_{xy}');
    ylabel('K_{zz}');
end
end

%dimensions changed to be in order kxy, then
%rows are angle and columns are kzz
disp('Plotting k_{meas}/k_{xy} vs. \theta and k_{z}/k_{xy}...');
kms=cell(size(ks));
for i=1:length(angles)
    kmsbyangle = cellfun(@(prison) prison{1}, answers{i});
    for j=1:length(ks)
        %Normalize by particular kxy
        kms{j} = [kms{j}; kmsbyangle(:,j)'/ks(j)];
    end
end
end
figure;
hold on;
[kgrid, anggrid] = meshgrid(ks, angles);
for n=1:length(ks)
    x = reshape(anggrid',[],1), reshape(kgrid'/ks(n),[],1)
    y = reshape(kms{n}',[],1)

```

```

    plot3(x, y, '*-', 'color', colores(n, length(ks)));
end
%legend(arrayfun(@(x) num2str(x), ks, 'UniformOutput', false));
xlabel('angle (degrees)');
ylabel('k_{zz}/k_{xy}');
zlabel('k_{meas}/k_{xy}');

```

B.8 tabulator.m

```

%tabulator
%turns lame structures into some csv action

%given params:
%  answers
%  angles
%  kxy
%  kz

% Bad style, but I'm dealing with a cluttered namespace
% because I'm not functionalizing these.
% This is because parameter passing is annoying. So, leave me alone.
[Kxy, Kz] = meshgrid(kxy, kz);

fprintf('angle, kxy, kz, kmeas \n');
for t=1:length(angles),
    for pt=1:size(Kxy,1)*size(Kxy,2),
        fprintf([ num2str(angles(t)) ', ' ...
                  num2str(Kxy(pt)) ', ' ...
                  num2str(Kz(pt)) ', ' ...
                  num2str(answers{t}{pt}{1}) '\n']);
    end
end

```

B.9 symmetric_tocell.m

```

function a=symmetric_tocell(A)
% Converts a symmetric matrix A into a cell

```

```

% containing the values, comsol-style.
% I need to make sure comsol likes them nx1 instead of 1xn.

%test for squareness
assert(size(A,1)==size(A,2), ...
    'Dawg that matrix aint square much less symmetric');
%test for symmetry
for i=1:size(A,1),
    for j=i:size(A,1),
        if (A(i,j)~=A(j,i)),
            disp(['Warning: Dawg that matrix aint square (A(' ...
                int2str(i) int2str(j) ')=' num2str(A(i,j)) ...
                ' != A(' int2str(j) int2str(i) ...
                ')=' num2str(A(j,i)) ' ! )']]);
        end
    end
end

%The actual heavy lifting.
a={};
for m=1:size(A,1),
    %Takes upper-triangular section of mth column
    for element=A(1:m,m)'
        a{size(a,1)+1,1}=element;
    end
end

end

function t=triangle(n)
    t=(n^2+n)/2;
end

```

Appendix C

Code Used for Chapter 4

C.1 testtools.py

“testtools.py” is a loose port of code written by Dr. Matthew Sturm for IGOR Pro to do a similar analysis.

```

from __future__ import division

import tablib
from math import pi

# csv.reader and tablib aren't smart enough to read in numbers as
# numbers. Therefore, I map this over the strings in the rows spat
# out by csv.reader.
def str2num(string):
    import re

    if re.match('^-\d*\.\d*$', string):
        return float(string)
    elif re.match('^-\d+$', string):
        return int(string)
    else:
        return string

def import_raw_data(filename):
    import csv

    data = tablib.Dataset()
    for row in csv.reader(open(filename, 'r')):
        data.append( map(str2num, row) )

    data.headers = ( 'logger_id'
                     , 'day'
                     , 'hourmin'

```

```

        , 'sec'
        , 'needletemp'
        , 'reftemp'
        , 'volts'
        , 'timer')

    return data

def unique(col):
    return list(set(col))

# Time is kept in two ints: one of the form hhmm, and the other in s.
# This function converts those to absolute seconds, and rebuilds the
# table appropriately.
def hms_to_s(data):
    # This bit here converts hhmm to seconds and adds to s.
    # It doesn't account for changes in the julian days.
    # Just don't test @ midnight, I guess.
    def convert(hourmin, sec):
        return 60*(hourmin%100) + sec + 3600*(hourmin//100 )

    def sieve(st):
        return (st != 'hourmin') and (st != 'sec')

    sec = map(lambda t: convert(t[0], t[1]),
              zip(data['hourmin'], data['sec']))

    new_data = zip(sec, *map(lambda h: data[h],
                             filter(sieve, data.headers) ))
    new_headers = ['sec']+filter(sieve, data.headers)
    return tablib.Dataset(*new_data, headers=new_headers)

def tab_filter(data, header, testfxn):

```

```

new_data = [];
for (i, pt) in enumerate(data[header]):
    if testfxn(pt):
        new_data.append(data[i])
return tablib.Dataset(*new_data, headers=data.headers)

def tab_pprint(data):
    width = 2 + max(map(lambda st: len(st), data.headers))

    def padder(st):
        l = len(str(st))
        return st + (width-l)*" " if l <= width else st[:width]

    print " | ".join(map(padder, data.headers))
    print "-"*((width+2)*len(data.headers)-1)
    for row in data:
        print " | ".join(map(padder, map(str, row)))

def tab_plot(data, x_header, y_headers = None, fit=None):
    import matplotlib.pyplot as pyplot

    headers = filter(lambda h: h != x_header, data.headers)
                if y_headers == None else y_headers)

    xs = [ data[x_header] for header in headers ]
    ys = [ data[header] for header in headers]

    if (fit != None):
        from numpy import arange, exp, log, polyval
        more_xs = arange(data[x_header][1], data[x_header][-1])
        more_ys = polyval(fit, log(more_xs))
        xs+=more_xs

```



```

        ys+= [more_ys]

    pyplot.semilogx(*reduce(lambda a, b: a+b, zip(xs,ys)))
    pyplot.xlabel(x_header)
    pyplot.ylabel('Everything Else')
    pyplot.show()
    return data

#Repeats code from hms_to_s, or whatever I called that fxn.
#Ideally, I would generalize the ideas of, "do something with
#these columns, generate THIS column" and "Get rid of these
#columns.
def relative_time(data, h_abs="sec", h_rel="sec"):
    from numpy import array

    t_actual = data[h_abs]
    t_zero = t_actual[0]

    t_relative = list(array(t_actual) - t_zero)

    def sieve(st):
        return (st != h_abs)

    new_data = zip(t_relative, *map(lambda h: data[h],
                                   filter(sieve, data.headers) ))

    new_headers = [h_rel]+filter(sieve, data.headers)

    return tablib.Dataset(*new_data, headers=new_headers)

#A class of tools for splitting data up
class Splitters(object):

```

```

#You change your mind like a girl changes clothes!
@staticmethod
def hot_and_cold(data):
    from numpy import array, floor
    from scipy.optimize import curve_fit

    def fit(x, a, b, c):
        y = map(lambda x: (b if (x-a) > 0 else 0) - c, x)
        return array(y)

    split = int(floor(curve_fit(fit,
                                data['sec'],
                                data['volts'],
                                ( 0.5*(data['sec'][0]+data['sec'][1]),
                                data['volts'][0],
                                0 ))[0][0]))

    hot = tablib.Dataset( *data[slice(None, split, None)],
                          headers=data.headers)
    cold = tablib.Dataset( *data[slice(split, None, None)],
                          headers=data.headers)

    return ( hot, cold )

@staticmethod
def manual(data, header, value):
    a = tab_filter(data, header, lambda x: x < value)
    b = tab_filter(data, header, lambda x: x >= value)
    return (a, b)

def linreg(data, xheader = 'sec', yheader = 'needletemp' ):
    from numpy import polyfit, log
    return polyfit(log(data[xheader][1:]), data[yheader][1:], 1)

```

```

def q(data):
    from numpy import average

    # I'm not sure what these constants mean, but based on the equation
    # I can make a pretty good guess!
    r_r = 10.6 #needle radius?
    r_h = 142.2 #resistance?
    l = 0.120 #needle length?

    volts = float(average(data['volts']))

    return ((volts/1000)**2.0 * r_h) / (l*r_r**2.0)

def heating_curve(data, q):
    const = linreg(data)[0]
    return q/4.0/pi/float(const)

def cooling_curve(cool_data, q):
    const = linreg(cool_data)[0]
    return -q/4/pi/float(const)

#applies mcgaw cooling curve. Untested.
def mcgaw(data, k_hot, q_hot, hot_period):
    from math import exp, log, pi

    correction = 4*pi*q_hot*k_hot*log( (exp(data['sec'][-1])+ hot_period) /
                                         hot_period)

    #print data['sec'][-1]
    #print hot_period

```

```

    #print correction

    newtemps = map( lambda x: x - correction, data['needletemp'])

    new_data = [];
    new_headers = data.headers

    for header in data.headers:
        if header == 'needletemp':
            #print "ding!"
            new_data.append(newtemps)
        else:
            new_data.append(data[header])

    #transposition!
    new_data = zip(*new_data)

    return tablib.Dataset(*new_data, headers=new_headers)

#Lachenbruch's Time Correction from a 1957 paper.
#Haven't been able to find said paper. Whatever.
def lachenbruch(data, dt):
    from numpy import exp, log

    data['needletemp'] = list(log(exp(data['needletemp']) - dt))

    return data

```

Appendix D
Raw Results of Numerical Model

Table D.1: $\theta = [0 : 5 : 90]$, $[k_{xy}, k_z] = [0.3, 0.5]$

Angle (degrees)	k_{xy}	k_z	k_{meas}
0	0.3	0.5	0.42503
5	0.3	0.5	0.42439
10	0.3	0.5	0.42439
15	0.3	0.5	0.41927
20	0.3	0.5	0.41927
25	0.3	0.5	0.41927
30	0.3	0.5	0.40758
35	0.3	0.5	0.40047
40	0.3	0.5	0.39267
45	0.3	0.5	0.38462
50	0.3	0.5	0.37641
55	0.3	0.5	0.37196
60	0.3	0.5	0.36418
65	0.3	0.5	0.36418
70	0.3	0.5	0.35754
75	0.3	0.5	0.35574
80	0.3	0.5	0.35574
85	0.3	0.5	0.35574
90	0.3	0.5	0.3589

Table D.2: $\theta = 0$, $[k_{xy}, k_z] = \text{meshgrid}(\text{linspace}(0.2, 0.4, 4), \text{linspace}(0.2, 0.4, 4))$

Angle (degrees)	k_{xy}	k_z	k_{meas}
0	0.2	0.2	0.21614
0	0.2	0.26667	0.2492
0	0.2	0.33333	0.28009
0	0.2	0.4	0.30801
0	0.26667	0.2	0.25315
0	0.26667	0.26667	0.29095
0	0.26667	0.33333	0.32668
0	0.26667	0.4	0.35519
0	0.33333	0.2	0.28594
0	0.33333	0.26667	0.32836
0	0.33333	0.33333	0.36839
0	0.33333	0.4	0.40493
0	0.4	0.2	0.31911
0	0.4	0.26667	0.36629
0	0.4	0.33333	0.40706
0	0.4	0.4	0.44717

Table D.3: $\theta = [30 : 15 : 90]$, $k_{xy} = 0.3$, $k_z = [0.1 : 0.1 : 0.5]$

Angle (degrees)	k_{xy}	k_z	k_{meas}
30	0.3	0.1	0.23
30	0.3	0.2	0.28322
30	0.3	0.3	0.3296
30	0.3	0.4	0.3704
30	0.3	0.5	0.40758
45	0.3	0.1	0.26424
45	0.3	0.2	0.29853
45	0.3	0.3	0.3296
45	0.3	0.4	0.35806
45	0.3	0.5	0.38462
60	0.3	0.1	0.29307
60	0.3	0.2	0.31317
60	0.3	0.3	0.3296
60	0.3	0.4	0.34888
60	0.3	0.5	0.36418
75	0.3	0.1	0.31449
75	0.3	0.2	0.32006
75	0.3	0.3	0.3296
75	0.3	0.4	0.33925
75	0.3	0.5	0.35574
90	0.3	0.1	0.32201
90	0.3	0.2	0.32375
90	0.3	0.3	0.3296
90	0.3	0.4	0.33566
90	0.3	0.5	0.3589

Table D.4: $\theta = [0 : 30 : 90]$, $k_{xy} = 0.3$, $k_z = [0.1, 0.5]$

Angle (degrees)	k_{xy}	k_z	k_{meas}
0	0.3	0.1	0.19287
0	0.3	0.5	0.42503
30	0.3	0.1	0.23
30	0.3	0.5	0.40758
60	0.3	0.1	0.29307
60	0.3	0.5	0.36418
90	0.3	0.1	0.32201
90	0.3	0.5	0.3589

Table D.5: $\theta = [5 : 5 : 15]$, $k_{xy} = 0.3$, $k_z = [0.1, 0.5]$

Angle (degrees)	k_{xy}	k_z	k_{meas}
5	0.3	0.1	0.19429
5	0.3	0.5	0.42439
10	0.3	0.1	0.19643
10	0.3	0.5	0.42247
15	0.3	0.1	0.20301
15	0.3	0.5	0.41927

Appendix E

Needle Probe Apparatus Directions

E.1 Taking A Measurement

The apparatus is controlled using a keypad that enables one to communicate with the data logger using Campbell “star-codes.” Initiating a test is a matter of clearing and setting some registers using these star-codes.

In general, the apparatus is used like so:

1. Turn on the device.
2. Insert the needle into medium being measured.
3. Use star-codes to clear the first three registers. “* 6 A” accesses the registers, and “D *n*” toggles the *n*th register. For example, to clear the second register, press “* 6 A D 2” .
4. Turn on the first register by pressing “* 6 A D 1”. This makes the apparatus measure temperature.
5. Turn on the second register by pressing “* 6 A D 2”. This turns on the heating element, effectively starting the test.
6. Wait 20 minutes for test to complete.

E.2 CSV Headers

Data from the Campbell instrument comes in the form of a .csv file. For this particular experiment, the columns (from left to right) represent:

1. An instrument ID (constant in this case).
2. Ordinal day, out of 366. For example, March 17th is day 76.
3. hh:mm portion of timestamp. For example, 6:30pm is represented as 1830.
4. Seconds portion of timestamp.
5. Needle temperature, in Celcius.

6. Reference temperature, in Celcius.
7. Voltage across needle probe, in millivolts.
8. Experiment timer, in seconds.